Problèmes de discrétisation et de filtrage pour la visualisation d'images numériques

Djamchid Ghazanfarpour-Kholendjany

Ecole Nationale Supérieure des Mines de Saint-Etienne

Université de Saint-Etienne

N° d'ordre : 26 ID

THESE DE DOCTORAT

présentée par

Djamchid GHAZANFARPOUR-KHOLENDJANY

(DOCTEUR INGENIEUR en INFORMATIQUE)

pour obtenir le titre de

DOCTEUR

de l'Université de Saint-Etienne

et de l'Ecole Nationale Supérieure des Mines de Saint-Etienne

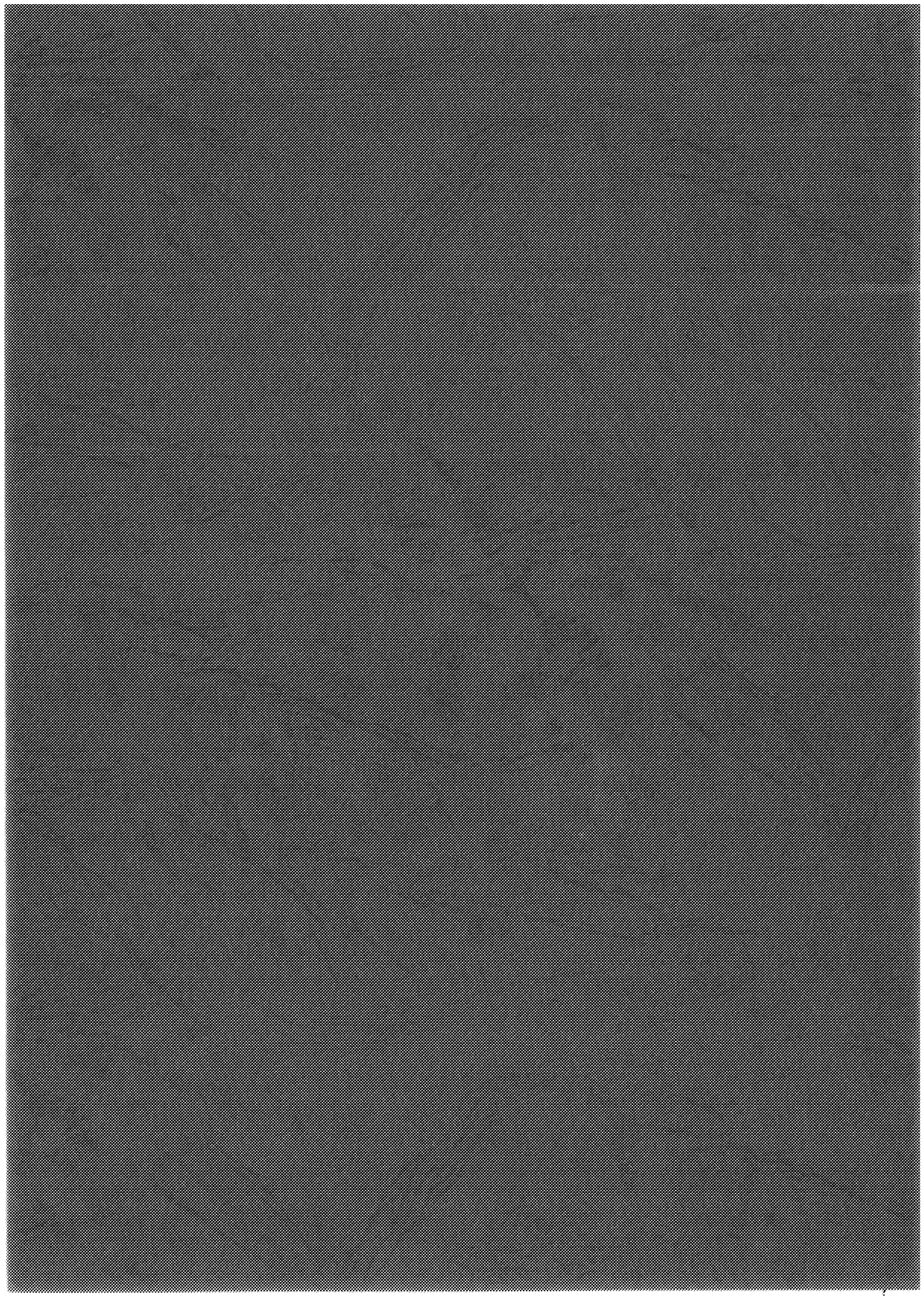
Spécialité : Informatique, Image, Intelligence Artificielle et Algorithmique

PROBLEMES DE DISCRETISATION ET DE FILTRAGE POUR LA VISUALISATION D'IMAGES NUMERIQUES

Soutenue à Saint-Etienne le 17 Mai 1990

Composition du jury

Président et Rapporteur :	Monsieur	J. FRANCON
Rapporteur :	Monsieur	R. GOUTTE
Examinateurs :	Messieurs	J.M. CHASSERY
		B. PEROCHE
		P. ROUGNY



Ecole Nationale Supérieure des Mines de Saint-Etienne

Université de Saint-Etienne

N° d'ordre : 26 ID

THESE DE DOCTORAT

présentée par

Djamchid GHAZANFARPOUR-KHOLENDJANY

(DOCTEUR-INGENIEUR en INFORMATIQUE)

pour obtenir le titre de

DOCTEUR

de l'Université de Saint-Etienne

et de l'Ecole Nationale Supérieure des Mines de Saint-Etienne

***Spécialité** : Informatique, Image, Intelligence Artificielle et Algorithmique*

PROBLEMES DE DISCRETISATION ET DE FILTRAGE POUR LA VISUALISATION D'IMAGES NUMERIQUES

Soutenue à Saint-Etienne le 17 Mai 1990

Composition du jury

Président et Rapporteur :	Monsieur	J. FRANÇON
Rapporteur :	Monsieur	R. GOUTTE
Examinateurs :	Messieurs	J.M. CHASSERY
		B. PEROCHE
		R. ROUGNY

Remerciements

Je tiens à remercier :

Monsieur J.FRANÇON, Professeur à l'Université Louis Pasteur de Strasbourg, rapporteur, qui m'a fait l'honneur de présider le jury de cette thèse. Je le remercie également pour l'intérêt qu'il a manifesté à l'égard de mes activités de recherche depuis mon arrivée au Département d'Informatique de l'Université Louis Pasteur.

Monsieur R. GOUTTE, Professeur à l'Institut National des Sciences Appliquées de Lyon, rapporteur, qui a bien voulu examiner cette thèse.

Monsieur J.M. CHASSERY, Directeur de Recherche au Centre National de la Recherche Scientifique, et Monsieur R. ROUGNY, Professeur à l'Université de Saint-Etienne, pour l'intérêt qu'ils ont manifesté à l'égard de mes travaux et pour leur participation au jury de cette thèse .

Monsieur B. PEROCHE, Professeur à l'Ecole National Supérieure des Mines de Saint-Etienne, qui s'est intéressé de près depuis plusieurs années à mes travaux de recherche à l'occasion de la préparation de deux thèses et même après mon départ de Saint-Etienne. Ses remarques et ses encouragements m'ont été toujours précieux. Je le remercie également pour son amitié.

Tous mes amis collègues, personnes présentes et passées, des Départements d'Informatique de l'Ecole des Mines et de l'Université Louis Pasteur pour leur amitié et leur aide.

Ma famille pour ses encouragements constants et son soutien très précieux depuis toujours..

Table des matières

Chapitre III

Aliassage et antialiassage d'objets bi ou tridimensionnels

Chapitre IV

Antialiassage et tampon de profondeur

INTRODUCTION

Introduction

Les images de synthèse sont très présentes dans beaucoup de domaines tels que la C.A.O., l'E.A.O., la simulation, l'audiovisuel, le graphique d'affaires et la publicité. L'utilisation des différentes techniques de rendu (textures, éclairement, ombres portées, pénombre, transparence, ...) permet d'atteindre un plus grand degré de réalisme indispensable pour la plupart des domaines d'application des images de synthèse. Cependant, les images de synthèse ne peuvent pas être considérées comme réalistes tant que les différents défauts visibles dus à la discrétisation de la scène lors de son affichage existent. En effet, des contraintes technologiques imposent une faible définition aux images numériques. Il en résulte, dans la plupart des cas, un échantillonnage insuffisant de la scène et par conséquence les problèmes de repliement du spectre appelé aliassage ("aliasing"). Il faut donc résoudre les problèmes de passage d'une scène sous sa forme analogique à une image numérique avant son affichage.

Les problèmes de passage du continu au discret sont bien connus en traitement du signal dans le cas monodimensionnel. Les travaux de Catmull et de Crow, dans les années soixante dix, sont les premières études sérieuses sur ce sujet dans le domaine de la synthèse d'images numériques. Depuis, le sujet a été étudié par les chercheurs mais il reste toujours un domaine actif de recherche qui a souvent évolué en fonction des nouvelles contraintes imposées par les domaines d'application et par les nouvelles techniques de rendu utilisées en synthèse d'images.

On peut remarquer que le phénomène d'aliassage en synthèse d'images provoqué par la basse définition des mémoires de trames (mémoires d'images numériques) se manifeste essentiellement sous les trois formes suivantes :

1- La présence de marches d'escalier irrégulières ("jaggies") sur la frontière entre deux entités de couleurs différentes de la scène (cf. images III.1 et III.2).

2- L' apparition et la disparition ("flash") des petits objets suivant leur position dans la scène d'une séquence à l'autre pour les images animées, ou leur affichage en

pointillé dans le cas des petits objets allongés (cf. chapitre III) pour les images fixes ou dynamiques.

3- La présence d'anomalies dans des images contenant des textures, essentiellement sous la forme de moirés en particulier après application de transformations géométriques sur les textures (cf. image V.2).

Les méthodes d'antialiassage permettent de franchir un des obstacles majeurs à une bonne qualité et un grand réalisme nécessaires aux domaines d'applications de la synthèse d'images. Les problèmes de passage du continu au discret en synthèse d'images peuvent être étudiés de beaucoup de façons avec différentes possibilités de solutions théoriques ou pratiques. Dans les différents chapitres de cette thèse, nous étudions ce sujet ; dans un premier temps, nous l'abordons d'un point de vue théorique, puis nous en tirons les solutions pratiques pour la synthèse d'images. En particulier, nous proposerons plusieurs études et méthodes originales pour les cas les plus fréquents.

Dans le premier chapitre, nous étudions les problème de la discrétisation des signaux. Dans une étude théorique, nous généralisons l'échantillonnage régulier, l'aliassage et l'antialiassage au cas des signaux bidimensionnels (un rappel du cas monodimensionnel est présenté dans l'annexe A). La fin de ce premier chapitre est consacrée à l'étude de la méthode d'échantillonnage stochastiques qui est un nouvelle technique d'antialiassage employée en synthèse d'images.

Le deuxième chapitre présente essentiellement l'aliassage et l'antialiassage d'images dans un cadre général. En s'appuyant sur les bases théoriques du chapitre précédent, nous étudions les méthodes envisageables permettant de résoudre les problèmes de passage d'une scène sous sa forme analogique à une image numérique. D'autre part, nous définissons quelques notions de base, nous étudions les contraintes technologiques et nous présentons le rôle important joué par le système visuel sur la perception des problèmes d'aliassage.

Le troisième chapitre a trait à des problèmes d'aliassage et antialiassage d'objets bi ou tridimensionnels. Après avoir présenté les problèmes de marche d'escalier et des petits objets, nous classifions et étudions les diverses méthodes envisageables dans le cas des objets bidimensionnels. Dans la deuxième partie de ce chapitre, nous étudions les méthodes d'antialiassage 3D et les problèmes de leur application dans le cas des quatre méthodes les plus courantes d'éliminations de parties cachées des scènes en synthèse d'images.

Etant donnée l'importance de la méthodes du tampon de profondeur parmi les techniques d'élimination de parties cachées des scènes 3D et les difficultés importantes d'application d'une méthode d'antialiassage dans ce cas, le quatrième chapitre est consacré à ce sujet. Après avoir présenté les principaux problèmes d'antialiassage avec le tampon de profondeur, nous étudions brièvement les techniques existantes et nous proposons deux méthodes originales.

Le cinquième chapitre est consacré à la présence de textures et de moirés dans les images de synthèse. Dans un premier temps nous étudions les problèmes de transformation perspective des textures planes. Après avoir formalisé ce problème dans les deux domaines spatial et fréquentiel, nous classifions les diverses méthodes de traitement et nous présentons des nouvelles méthodes utilisant la transformation perspective directe ou inverse. Finalement, nous étudions brièvement l'emploi des textures volumiques en synthèse d'images.

CHAPITRE I

Discrétisation des signaux : Echantillonnages régulier et stochastique

CHAPITRE I

Discrétisation des signaux : Echantillonnages régulier et stochastique

I.1 Introduction

Le phénomène d'aliassage en synthèse d'images numériques est dû à un échantillonnage insuffisant de la scène (sous sa forme analogique) imposé par des contraintes technologiques. Une bonne connaissance du théorème d'échantillonnage et de restitution des signaux analogiques est indispensable afin d'expliquer l'origine de l'aliassage et de trouver des techniques pour résoudre ce problème en synthèse d'images. En effet, le phénomène d'aliassage est bien connu en théorie du signal ; c'est le repliement (recouvrement) du spectre fréquentiel (cf. annexe A). Cependant, il a fallu plusieurs années pour que les chercheurs du domaine graphique établissent une corrélation étroite entre les résultats de la théorie du signal et les défauts constatés sur les images de synthèse. Les travaux présentés dans [CATM 74], approfondis et généralisés par ceux présentés dans [CROW 76] et [BLIN 76], sont les premières études que l'on peut citer au sujet de l'aliassage dans le domaine de la synthèse d'images numériques.

Ce chapitre est étroitement lié à l'annexe A dans lequel nous avons étudié l'échantillonnage régulier et le phénomène d'aliassage pour les signaux monodimensionnels afin de clarifier le problème dans le cas le plus simple, de définir l'origine du problème et de présenter les méthodes utilisées pour le résoudre. Dans ce chapitre, nous généraliserons cette étude au cas des signaux bidimensionnels qui nous intéressent plus particulièrement (l'image étant un signal bidimensionnel). Nous terminerons ce chapitre par une étude sur la technique d'antialiassage par l'échantillonnage stochastique des signaux.

I.2 Echantillonnage régulier et reconstitution d'un signal bidimensionnel

Les résultats obtenus pour l'échantillonnage régulier d'un signal monodimensionnel (cf. Annexe A) se généralisent au cas des signaux bidimensionnels [KUNT 80]. Etant donné

qu'une image numérique est un signal discret spatial bidimensionnel, cette étude nous permet d'expliquer l'origine des phénomènes d'aliassage en synthèse d'images.

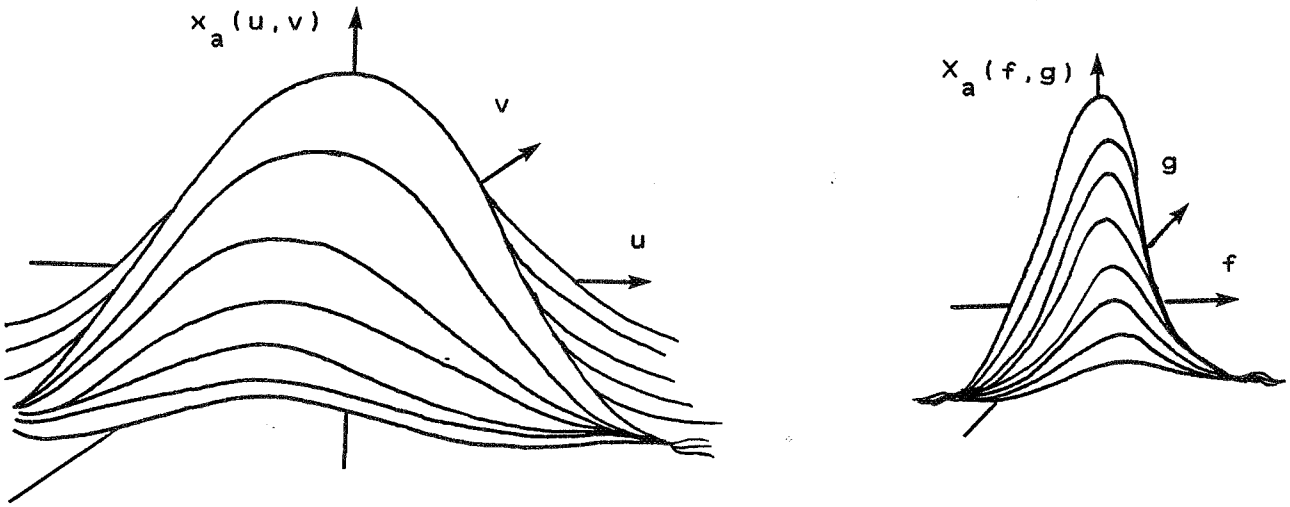
Soit $x_a(u,v)$ un signal analogique bidimensionnel à support non-borné dans le domaine spatial (cf.figure I.1.a). La transformée de Fourier bidimensionnelle de $x_a(u,v)$, qui définit le spectre du signal, est :

$$X_a(f,g)= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_a(u,v).e^{-j2\pi.(f.u+g.v)}.du.dv \qquad (I.1)$$

où f et g sont les fréquences spatiales en cycle par unité de distance. $X_a(f,g)$ est un signal dont les largeurs de bandes sont limitées (cf.figure I.1.b) :

$$X_a(f,g)= 0 \quad si \quad |f|>f_{max} \text{ ou } |g|>g_{max} \qquad (I.2)$$

où f_{max} et g_{max} sont les fréquences maximales du signal bidimensionnel.



(a) Dans le domaine spatial. (b)Dans le domaine fréquentiel.

Figure I.1 : Représentations spatiale et fréquentielle d'un signal analogique bidimensionnel.

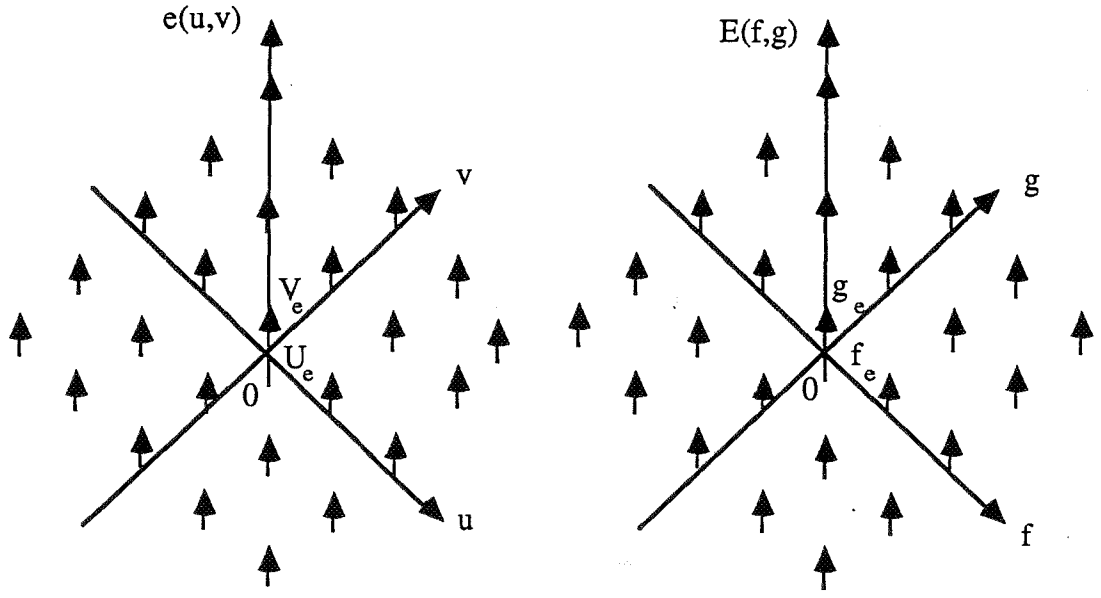
Le signal d'échantillonnage e(u,v) est une suite périodique et bidimensionnelle d'impulsions de Dirac (cf.figure I.2.a) donnée par l'expression :

$$e(u,v)= \sum_{m=-\infty}^{m=+\infty} \sum_{n=-\infty}^{n=+\infty} \delta(u-m.U_e,v-n.V_e) \qquad (I.3)$$

où U_e (resp. V_e) est la période d'échantillonnage selon u (resp. v). La transformée de Fourier bidimensionnelle de e(u,v) définit le spectre du signal d'échantillonnage (cf. figure I.2.b).

$$E(f,g)= f_e.g_e \sum_{m=-\infty}^{m=+\infty} \sum_{n=-\infty}^{n=+\infty} \delta(f-m.f_e,g-n.g_e) \qquad (I.4)$$

où $f_e=\frac{1}{U_e}$ (resp. $g_e=\frac{1}{V_e}$) est la fréquence d'échantillonnage selon f (resp. g).



(a) Dans le domaine spatial. (b)Dans le domaine fréquentiel.

Figure I.2 : Représentations spatiale et fréquentielle du signal d'échantillonnage bidimensionnel.

Comme dans le cas d'un signal monodimensionnel (cf. annexe A), le signal échantillonné bidimensionnel est défini par un produit ordinaire entre le signal analogique et le signal d'échantillonnage dans le domaine spatial (cf.figure I.3.a) :

$$x_e(u,v)= x_a(u,v).e(u,v)$$

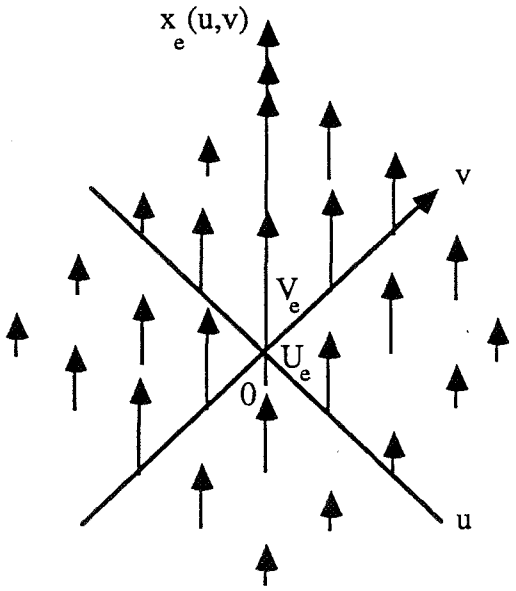
$$x_e(u,v)=x_a(u,v) \sum_{m=-\infty}^{m=+\infty} \sum_{n=-\infty}^{n=+\infty} \delta(u-m.U_e,v-n.V_e) \qquad (I.5)$$

ou par un produit de convolution bidimensionnel, noté **, dans le domaine fréquentiel :

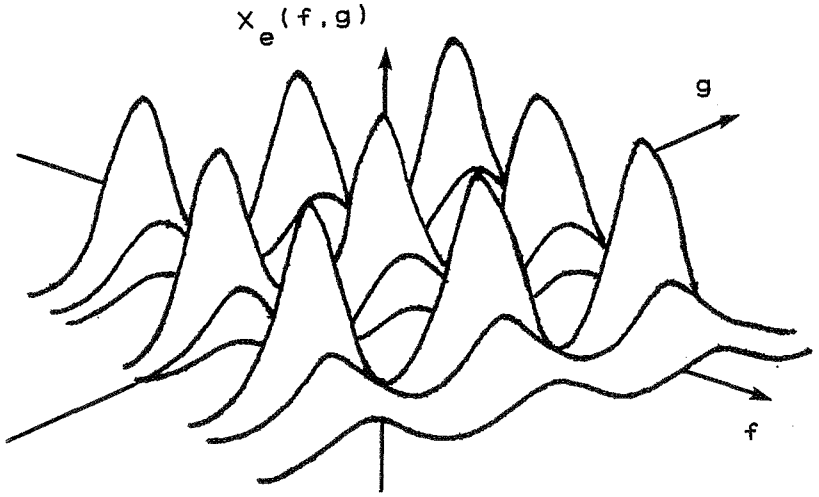
$$X_e(f,g)= X_a(f,g)**E(f,g)$$

$$X_e(f,g)= f_e.g_e \sum_{m=-\infty}^{m=+\infty} \sum_{n=-\infty}^{n=+\infty} X_a(f-m.f_e,g-n.g_e) \qquad (I.6)$$

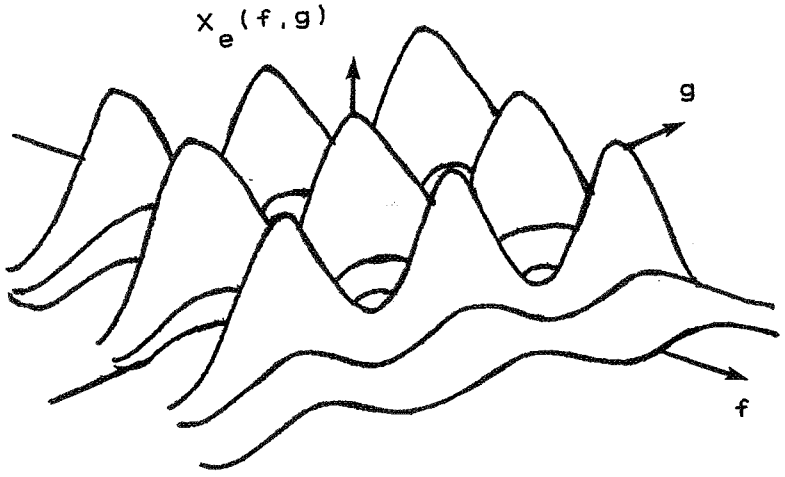
On constate que $X_e(f,g)$ est un signal périodique dans le domaine fréquentiel avec des périodes égales à f_e et g_e selon les directions de f et de g. Comme dans le cas monodimensionnel, on distingue essentiellement deux cas d'échantillonnage, avec ou sans repliement du spectre (cf. figures I.3.b et I.3.c).



(a) Dans le domaine spatial.



(b) Dans le domaine fréquentiel , sans repliement du spectre.



(c) Dans le domaine fréquentiel , avec repliement du spectre.

Figure I.3 :Représentation du signal échantillonné bidimensionnel.

1- Echantillonnage bidimensionnel sans repliement de spectre :

$$\text{si : } f_e \geq 2.f_{max} \text{ et } g_e \geq 2.g_{max} \tag{I.7}$$

alors il n'y a pas de repliement du spectre (cf.figure I.3.b). La reconstitution fidèle du signal analogique d'origine, $x_a(u,v)$, est réalisable à partir de ses échantillons, à l'aide d'un filtre passe-bas bidimensionnel idéal. Le filtrage garde le motif central du spectre en supprimant tous les autres motifs. Dans le domaine fréquentiel nous avons :

$$X_a(f,g)=X_e(f,g).H(f,g) \tag{I.8}$$

où H(f,g), est la fonction de transfert du filtre de reconstitution (cf. figure I.4.b).

$$H(f,g) = \begin{cases} \dfrac{1}{f_e.g_e} & \text{si} \quad |f| \leq \dfrac{f_e}{2} \text{ et } |g| \leq \dfrac{g_e}{2} \\ 0 & \text{si} \quad |f| > \dfrac{f_e}{2} \text{ ou } |g| > \dfrac{g_e}{2} \end{cases} \tag{I.9}$$

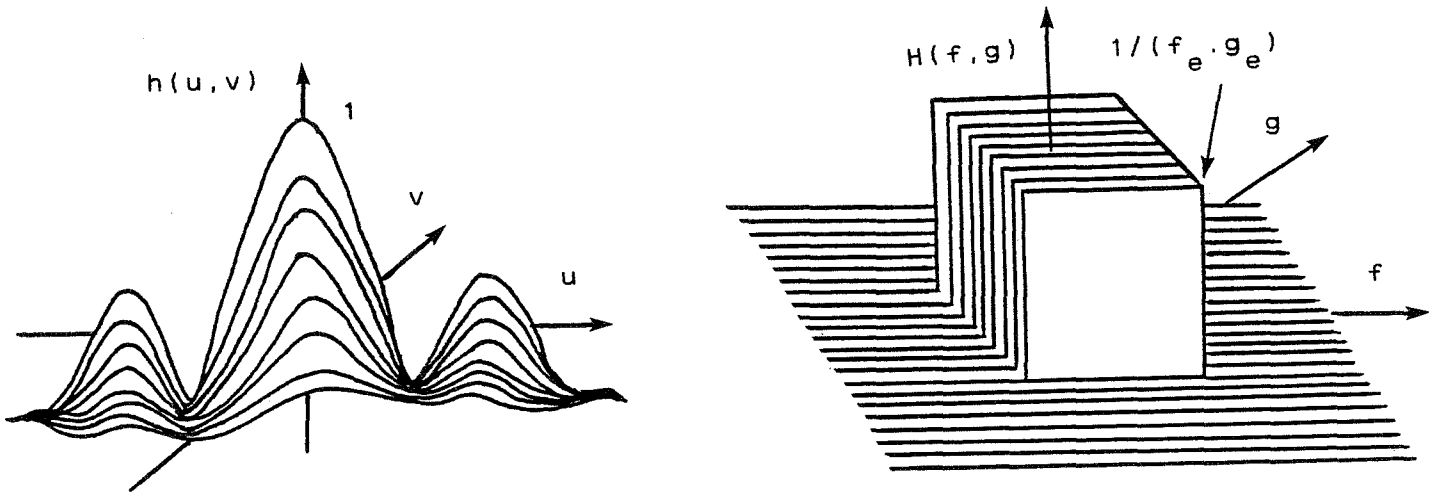
Dans le domaine spatial, le filtrage est réalisé par un produit de convolution bidimensionnel entre $x_a(u,v)$ et h(u,v) qui est la réponse impulsionnelle (cf.figure I.4.a) du filtre passe-bas de reconstitution :

$$x_a(u,v)= x_e(u,v)**h(u,v) \tag{I.10}$$

Comme précédemment, h(u,v) est la transformée de Fourier inverse bidimensionnelle de H(f,g) :

$$h(u,v)= \frac{\mathrm{Sin}\left(\pi.\frac{u}{U_e}\right)}{\pi.\frac{u}{U_e}} . \frac{\mathrm{Sin}\left(\pi.\frac{v}{V_e}\right)}{\pi.\frac{v}{V_e}}$$

$$h(u,v)= \mathrm{Sinc}\left(\frac{u}{U_e}, \frac{v}{V_e}\right) \tag{I.11}$$



(a) Réponse impulsionnelle. (b) Fonction de transfert.

Figure I.4 : Filtre passe-bas bidimensionnel idéal de reconstitution.

2- Echantillonnage bidimensionnel avec repliement du spectre :

On peut distinguer trois types de repliement du spectre du signal échantillonné bidimensionnel :

- si $f_e<2.f_{max}$ et $g_e\geq 2.g_{max}$ (I.12)
  il y a un repliement du spectre dû à un sous-échantillonnage suivant la direction de u.

- si $f_e\geq 2.f_{max}$ et $g_e<2.g_{max}$ (I.13)
  il existe un repliement du spectre dû à un sous-échantillonnage suivant la direction de v.

- si $f_e<2.f_{max}$ et $g_e<2.g_{max}$ (I.14)
  il y a un repliement du spectre dû aux sous-échantillonnages selon les deux directions (cf. figure I.3.c).

Il est à remarquer que dans le cas bidimensionnel, le repliement du spectre donne lieu à trois cas distincts alors que l'on n'en trouve qu'un dans le cas monodimensionnel (cf. annexe A). Il est important de distinguer ces trois cas de repliement du spectre, pour pouvoir appliquer un traitement d'antialiassage correct. Nous verrons par la suite les problèmes posés par un traitement effectué sans tenir compte de cette distinction (cf. chapitre V).

I.3 Antialiassage bidimensionnel dans le cas d'un échantillonnage régulier

Comme on l'a indiqué pour l'échantillonnage régulier monodimensionnel (cf. annexe A), le traitement d'antialiassage dans le cas d'un échantillonnage bidimensionnel régulier consiste essentiellement à adopter une des deux solutions suivantes :

1- augmentation des fréquences d'échantillonnage ;

2- limitation du spectre fréquentiel du signal bidimensionnel.

Pour les mêmes raisons que dans le cas monodimensionnel, la deuxième solution est plus réaliste. Elle consiste à utiliser un filtre passe-bas bidimensionnel (filtre antirepliement) pour limiter le spectre $X'_a(f,g)$ du signal analogique avant son échantillonnage.

$$X_a(f,g)= X'_a(f,g).H'(f,g) \tag{I.15}$$

où $X_a(f,g)$ est le spectre d'un signal bidimensionnel limité aux fréquences f_c et g_c si $H'(f,g)$ (la fonction de transfert du filtre antirepliement) est décrite par l'expression :

$$H'(f,g) = \begin{cases} 1 & \text{si } |f|\leq f_c \text{ et } |g|\leq g_c \\ 0 & \text{si } |f|>f_c \text{ ou } |g|>g_c \end{cases} \tag{I.16}$$

Dans le domaine spatial, le filtrage (produit de convolution bidimensionnel) s'exprime par :

$$x_a(u,v)=x'_a(u,v)**h'(u,v) \tag{I.17}$$

où $h'(u,v)$ est la réponse impulsionnelle du filtre antirepliement.

I.4 Echantillonnage stochastique et antialiassage

Nous avons vu précédemment que l'échantillonnage régulier avec une fréquence d'échantillonnage inférieure à deux fois la fréquence maximale du signal produit de l'aliassage. L'échantillonnage stochastique peut être utilisé afin d'éviter l'aliassage, mais en contrepartie, il génère du bruit. Pratiquement, un échantillonnage régulier peut conduire à un défaut régulier (aliassage) et un échantillonnage irrégulier (du type stochastique) peut conduire à un défaut irrégulier (du type bruit) qui peut être considéré comme moins important dans certains domaines d'applications (comme l'image de synthèse). Les méthodes d'antialiassage par échantillonnage stochastique qui sont déjà bien connues en

théorie du signal [SHAP 60] [LENE 66] [MASR 78] ont été introduites en synthèse d'image par [DIPP 85] et [COOK 86], en particulier dans le cas de la visualisation par la méthode de lancer de rayons (cf. chapitre III). L'échantillonnage stochastique peut diminuer de manière satisfaisante les défauts d'aliassage sur les images de synthèse au prix de l'ajout de bruits. Les expériences montrent que l'œil humain est moins sensible au bruit dû à un échantillonnage stochastique qu'à l'aliassage provoqué par un échantillonnage régulier insuffisant.

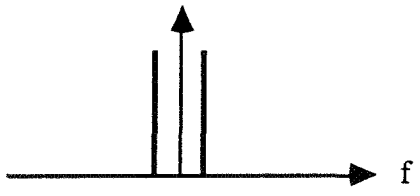
L'échantillonnage stochastique peut utiliser différents types de distributions aléatoires. Un type d'échantillonnage stochastique appelé poissonnien ("Poisson disk sampling") utilisant une distribution de Poisson, avec la restriction d'une distance minimale séparant deux échantillons, est bien adapté au cas de l'image. En fait, un tel échantillonnage se produit dans le système visuel humain (cf. chapitre II) grâce auquel et en complément avec le filtrage passe-bas effectué par le cristallin, l'aliassage peut être évité [YELL 83] [COOK 86].

Nous allons étudier l'échantillonnage stochastique en utilisant une distribution de Poisson. Pour simplifier le sujet, un signal monodimensionnel sinusoïdal X(f) pour une étude schématique (d'après [COOK 86]) est utilisé. La comparaison entre l'échantillonnage régulier de ce signal et son échantillonnage stochastique met en évidence l'efficacité de ce dernier pour éviter l'aliassage.

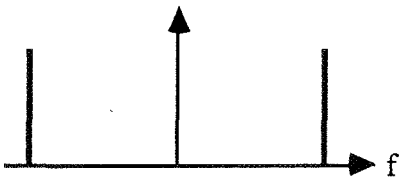
Dans un premier temps nous rappelons à l'aide de la figure I.5, l'échantillonnage régulier (cf. annexe A) de X(f) pour les deux cas suivants :

1- X(f) est un signal dont la fréquence est inférieure à la moitié de la fréquence d'échantillonnage (cf. figure I.5.a). Dans ce cas, X(f) est fidèlement restitué (cf. figure I.5.e).

2- X(f) est un signal dont la fréquence est supérieure à la moitié de la fréquence d'échantillonnage (cf. figure I.5.f). Dans ce cas, le signal restitué contient de l'aliassage (cf. figure I.5.j).

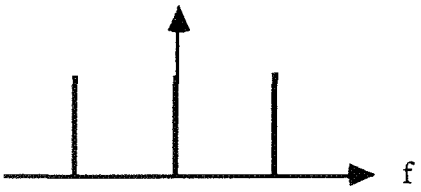


(a) signal original X(f)

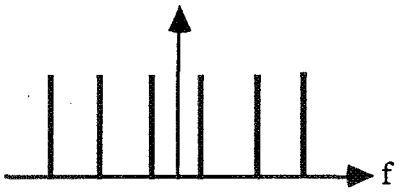


(f) signal original X(f)

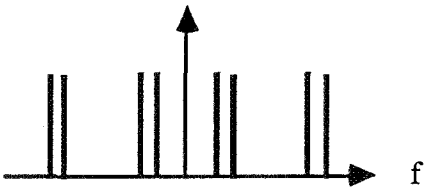
(b) signal d'échantillonnage Δ(f)



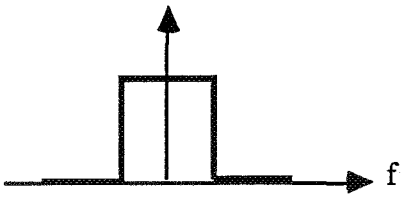
(g) signal d'échantillonnage Δ(f)



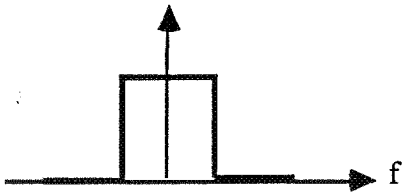
(c) signal échantillonné
X(f) ∗Δ(f)



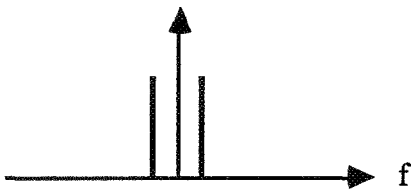
(h) signal échantillonné
X(f) ∗Δ(f)



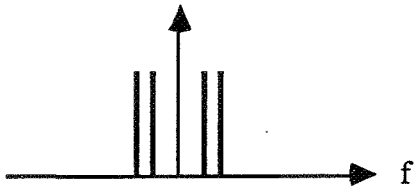
(d) filtre P.B. de restitution



(i) filtre P.B. de restitution



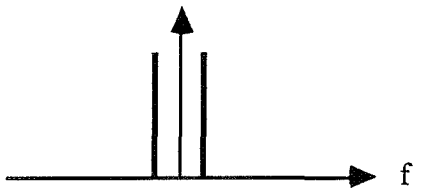
(e) signal restitué



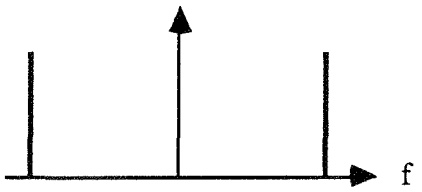
(j) signal restitué

Figure I.5 : Echantillonnage régulier d'un signal sinusoïdal d'après R. COOK.

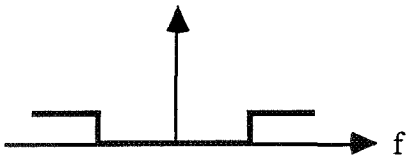
Après l'échantillonnage régulier de X(f), nous allons faire une étude schématique dans le domaine fréquentiel (cf. figure I.6) pour l'échantillonnage stochastique de ce signal dans les deux cas précédents. La fonction d'échantillonnage P(f) utilisée dans les deux cas est celle de Poisson. Le signal échantillonné (cf. figures I.6.c et I.6.h) est le résultat d'un produit de convolution entre le signal original X(f) et le signal d'échantillonnage P(f). Dans le premier cas, le signal original (cf. figure I.6.a) est fidèlement restitué (cf. figure I.6.e) à l'aide d'un filtre passe-bas de restitution. Le signal restitué dans le deuxième cas (cf. figure I.6.j) ne contient pas d'aliassage ; en fait, l'aliassage (cf. figure I.5.j) est remplacé par du bruit. L'amplitude du bruit décroît avec la restriction sur la distance minimale entre les échantillons [COOK 86].



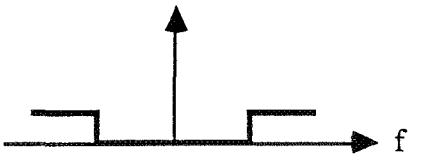
(a) signal original X(f)



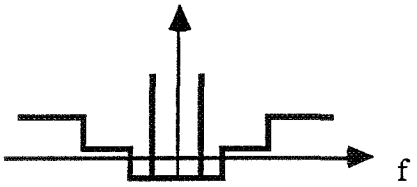
(f) signal original X(f)



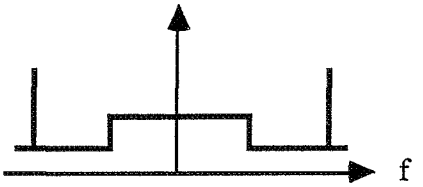
(b) signal d'échantillonnage P(f)



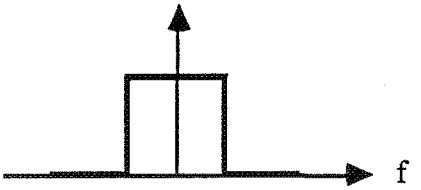
(g) signal d'échantillonnage P(f)



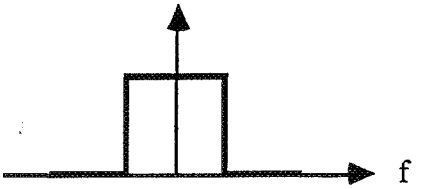
(c) signal échantillonné
X(f) * P(f)



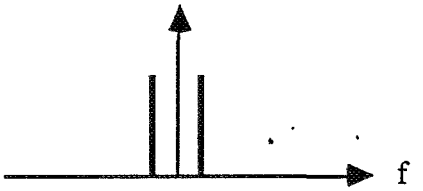
(h) signal échantillonné
X(f) * P(f)



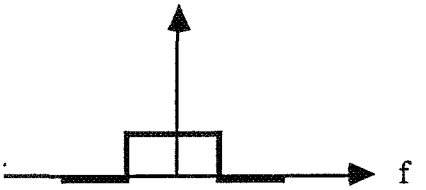
(d) filtre P.B. de restitution



(i) filtre P.B. de restitution



(e) signal restitué



(j) signal restitué

Figure I.6 : Echantillonnage stochastique d'un signal sinusoïdal par la fonction de Poisson d'après R.COOK.

L'echantillonnage stochastique par la distribution de Poisson est coûteux. La perturbation de la grille d'échantillonnage régulier par l'ajout de bruit ("jittering") peut approximer la distribution de Poisson. Cette méthode [SHAP 60] [BALA 62] [BROW 63] est la plus répandue pour un échantillonnage stochastique en particulier pour l'antialiassage en synthèse d'images [DIPP 85] [COOK 86].

I.5 Conclusion

L'échantillonnage inapproprié d'un signal bidimensionnel provoque de l'aliassage (repliement du spectre fréquentiel). L'antialiassage dans le cas d'un échantillonnage régulier peut être accompli des deux manière suivantes :

1- L'augmentation des fréquences d'échantillonnage afin qu'elles soient supérieures ou égales à deux fois les fréquences maximales du signal à échantillonner. Cette solution d'antialiassage est en général coûteuse et il existe des contraintes technologique pour son application dans le domaine de l'imagerie numérique.

2- La limitation du spectre fréquentiel du signal à la moitié des fréquences d'échantillonnage par un filtre passe-bas bidimensionnel antirepliement avant son échantillonnage. Cette solution est très souvent choisie dans la plupart des applications.

L'échantillonnage stochastique par une distribution aléatoire de type Poisson peut éviter l'aliassage (défaut de type régulier) en le remplaçant par du bruit (défaut de type irrégulier) qui est considéré comme un défaut moins perceptible par l'œil dans les images de synthèse.

Références du chapitre I

[BALA 62] : A. BALAKRISHNAN,
"On the Problem of Time Jitter in Sampling", IRE Transactions on Information Theory, April 1962, pp. 226-236.

[BLIN 76] : J. BLINN and M. NEWELL,
"Texture and Reflection in Computer Generated Images", Communications of the ACM, vol. 19, No. 10, October 1976, pp. 542-547.

[BROW 63] : W. BROWN,
"Sampling with Random Jitter", Journal Society Industrial Applied Mathematics, vol. 11, No. 2, June 1963, pp. 460-473.

[CATM 74] : E. CATMULL,
"A Subdivision Algorithm for Computer Display Curved Surfaces", Ph.D. thesis, University of Utah, December 1974, pp. 40-49.

[COOK 86] : R. COOK,
"Stochastic Sampling in Computer Graphics", ACM Transactions on Graphics, vol. 5, No. 1, January 1986, pp. 51-72.

[CROW 76] : F. CROW,
"The Aliasing Problem in Computer-Synthesized Shaded Images", Ph.D. Thesis, University of Utah, March 1976.

[DIPP 85] : M. DIPPE and E.WOLD,
"Antialiasing Through Stochastic Sampling", Computer Graphics, vol. 19, No. 3, July 1985, pp. 69-78.

[KUNT 80] : M. KUNT,
Traité d'électricité, vol. XX : Traitement numérique des signaux, Edition Georgi, 1980, pp. 329-342 et 349-355.

[LENE 66] : O. LENEMAN,
"Random Sampling of Random Processes : Impulse Response", Information and Control, vol. 9, 1966, pp. 347-363.

[MASR 78] : E. MASRY,
"Alias-Free Sampling : An Alternative Conceptualization and Its Applications", IEEE Transactions on Information Theory, vol. 24, No. 3, May 1978, pp. 317-324.

[SHAP 60] : H. SHAPIRO and R.SILVERMAN,
"Alias-free Sampling of Random Noise", SIAM J., vol. 8,No. 2, June 1960, pp. 225-248.

[YELL 83] : J. YELLOTT,
"Spectral Consequences of Photorecepteur Sampling in the Rhesus Retina", Science, vol. 221, July 1983, pp. 382-385.

CHAPITRE II

Aliassage et antialiassage d'images dans un cadre général

CHAPITRE II

Aliassage et antialiassage d'images dans un cadre général

II.1 Introduction

Dans le premier chapitre nous avons étudié l'origine du phénomène d'aliassage : l'insuffisance de l'échantillonnage. Dans le cas des images numériques, cette insuffisance est due la basse définition des mémoires de trame ("raster") existantes, imposée par des contraintes technologiques. Les défauts d'aliassage dans les images de synthèse peuvent être néanmoins supprimés ou diminués sensiblement grâce à des méthodes efficaces adaptées à chaque type de problème et à chaque type d'algorithme de visualisation.

Pour le chapitre actuel, nous nous limiterons à une étude générale sur l'aliassage et l'antialiassage des images de synthèse. Ce chapitre commence par quelques définitions sur les scènes, les images numériques et la grille d'affichage. Etant donnée la similitude entre les images de télévision et celles de synthèse obtenues sur une mémoire de trame, nous allons comparer les défauts d'aliassage pour ces deux types d'images afin de trouver une solution pour la synthèse d'images analogue à ce qui passe pour la télévision. Ensuite, une étude brève sur le rôle du système visuel humain et sa sensibilité à l'aliassage sera présentée. Finalement, nous analyserons les différents types de solutions théoriques ou pratiques envisageables pour l'antialiassage en synthèse d'images dans un cadre général avant de les détailler dans les chapitres suivants.

II.2 Scène, image numérique et mémoire de trame

Scène 3D :

On considère une scène 3D comme un ensemble d'objets définis dans un espace R^3 appelé le monde. Nous noterons (X,Y,Z) les coordonnées d'un point défini dans le repère du monde. En synthèse d'images, les objets de la scène 3D sont souvent modélisés par des entités géométriques telles que : segments, courbes, polygones, carreaux ("patchs") bilinéaires ou bicubiques.

Scène 2D :

Une scène 2D (image analogique) est décrite par des objets 2D qui sont soit directement définis dans un espace R^2, soit obtenus par la projection perspective d'une scène 3D sur l'espace R^2. L'espace R^2 dans lequel la scène 2D est définie est appelé l'espace écran. Nous noterons (x,y) les coordonnées d'un point défini dans le repère de l'écran. Nous considérons la scène (2D) comme une liste ordonnée d'objets 2D. Ces objets sont en général modélisés par des entités géométriques 2D. L'intensité (ou couleur) de ces objets peut être soit constante soit spatialement variable ; dans ce cas, l'objet contient une texture ou a une couleur dégradée. Par conséquent, une scène 2D est vue ici comme un signal bidimensionnel I(x,y) définissant l'intensité lumineuse et vérifiant la relation :

$$0 \leq I(x,y) \leq I_{max} \qquad \text{(II.1)}$$

où I(x,y) est un entier et I_{max} est l'intensité lumineuse maximale. Dans le cas d'une image couleur, elle est définie comme trois signaux R(x,y), V(x,y) et B(x,y) représentant les trois primaires rouge, verte et bleue.

Image numérique :

Différentes applications telles que l'affichage d'une scène ou son traitement supposent de calculer une image numérique (avec une définition donnée) à partir de la scène 2D. Dans le cas général, l'image numérique, I(i,j), est soumise à la relation suivante :

$$0 \leq I(i,j) \leq I_{max} \qquad \text{(II.2)}$$

où I(i,j) représente les échantillons de la scène aux points de coordonnées entières x=i, y=j. Les deux variables entières i et j sont finies et indépendantes. Il en résulte que l'image numérique est une matrice de points dont la définition dépend de i_{max} (resp. j_{max}) qui est le nombre d'échantillons suivant l'axe des x (resp. l'axe des y).

Mémoire de trame :

Nous verrons par la suite que l'image peut être calculée suivant diverses définitions selon les différents types de traitement qui lui seront appliqués. Quelle que soit cette définition, la phase ultime consiste à calculer l'image sur une grille correspondant à la définition d'affichage. C'est une grille régulière dont la distance entre deux points successifs est définie comme l'unité de distance. Chaque point de cette grille correspond au centre d'un pixel de la mémoire de trame (partie visualisée de la mémoire d'image). Les pixels sont définis comme des carrés de côté égal à l'unité (cf. figure II.1). Il faut remarquer que cette définition du pixel que nous avons choisie est le modèle le plus courant et qu'il existe d'autres définitions possibles comme des pixels hexagonaux ou des pixels circulaires. Par

ailleurs, il faut noter qu'un pixel sur l'écran (tube de rayons cathodiques) est physiquement constitué par plusieurs luminophores ayant pratiquement la même intensité lumineuse ou couleur.

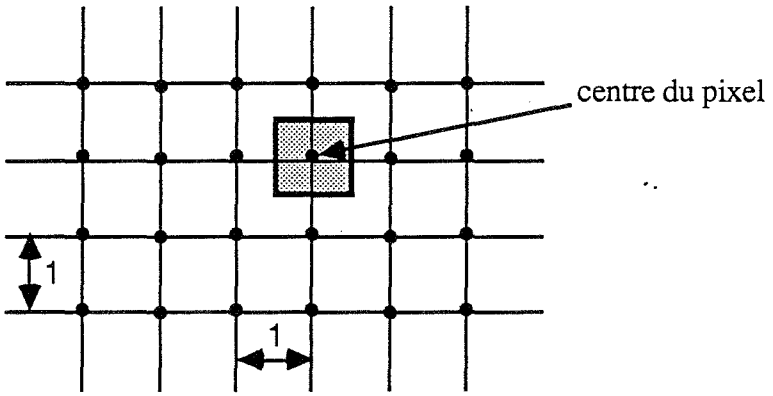


Figure II.1 : Grille d'affichage et définition d'un pixel.

L'un des paramètres importants d'une mémoire de trame est sa définition qui nous impose a priori, un nombre constant de pixels quelle que soit l'image à afficher. Une définition minimale est déterminée par le pouvoir de résolution de l'œil humain. Cette définition minimale doit rendre imperceptible à l'œil chacun des pixels de l'image affichée dans les conditions suivantes :

- affichage sur un moniteur de taille standard (de 9" à 23" [CONR 80]).
- distance de l'œil à l'image de l'ordre de trois fois la diagonale du moniteur (ceci n'est pas une règle absolue [CONR 80]).

Pour cette raison, la définition minimale des mémoires de trame utilisées est en général de l'ordre de 512x512, ce qui est voisin des standard des systèmes de télévision (souvent 525 ou 625 lignes, y compris les lignes de contrôle).

Le nombre de lignes utilisé pour représenter des images dans les systèmes de télévision est en général inférieur à celui utilisé pour des images de synthèse. Pourtant, les images issues de systèmes de télévision sont généralement perçues sans problèmes importants d'aliassage, ce qui n'est pas le cas des images issues d'une mémoire de trame. On peut donc se demander d'où parvient cette différence. Pour cela, nous allons comparer ces deux types d'images.

II.3 Comparaison des images de télévision et des images de synthèse

Les différences entre les images de télévision et les images de synthèse sont de deux ordres :

1- L'échantillonnage :

Dans un système à mémoire de trame, l'image est échantillonnée suivant les deux directions horizontale et verticale. Dans le cas de la télévision, l'image est échantillonnée par la caméra de télévision, uniquement suivant la direction verticale. Par conséquent, les problèmes d'échantillonnage, dans le cas d'une mémoire de trame, existent pour les deux directions tandis que ces problèmes sont dus uniquement à l'échantillonnage vertical dans le cas de la télévision.

2- L'effet du filtrage passe-bas de la caméra :

En pratique, le spot d'analyse utilisé lors de la prise de vue n'est pas ponctuel. En général, on peut le considérer comme une fonction ayant une base circulaire et une distribution qui peut être uniforme, gaussienne, en $\cos^2$, ... (cf. figure II.2). La dispersion du spot d'analyse joue un rôle important : en fait, quelle que soit la distribution du spot, il représente une surface finie de la scène, au lieu d'un seul point. Ceci se traduit par un filtrage passe-bas sur la scène (cf. annexe B). Ce résultat est extrêmement intéressant pour nous. Nous verrons par la suite que dans la plupart des cas les différentes méthodes d'antialiassage en synthèse d'images ne font que simuler les rôles d'échantillonnage et de filtrage de la caméra de télévision.

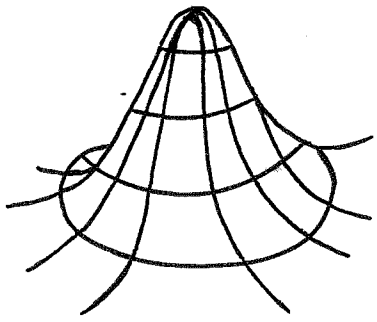


Figure II.2 : Un spot d'analyse réel (non ponctuel).

Si le filtrage passe-bas de la caméra est important (manque de mise au point ou "focus"), la scène peut être échantillonnée sans qu'il y ait de phénomène de repliement du spectre. En général, avec un "focus" normal, les images de télévision ne présentent pas de

défauts d'aliassage. Mais dans les scènes où il y a de forts changements d'intensité, le phénomène d'aliassage se présente souvent sous la forme de moiré (par exemple, un costume rayé) ou de marches d'escalier sur les segments, suivant des directions proches de l'horizontale (par exemple, la ligne de touche des terrains de sport).

La saisie d'image par une caméra vidéo suivie d'un échantillonnage de chaque ligne donne des résultats similaires à ce qu'on peut constater pour la télévision (cf. annexe B), malgré l'échantillonnage dans les deux directions. L'image II.1 est une saisie binaire par une caméra ; elle présente des marches d'escalier sur les bords (une saisie binaire par caméra équivaut à l'utilisation d'un spot ponctuel pour une scène bicolore). La même scène est saisie par caméra en 256 niveaux d'intensités différentes (cf. Image II.2), sans effet d'aliassage. L'effet du filtrage passe-bas de la caméra est évident sur l'image II.3 qui est le zoom x5 de l'image II.2 ; la dégradation des niveaux d'intensité entre le noir et le blanc est due au filtrage. Il faut noter que même dans le cas très particulier d'une saisie par caméra où l'image ne présente pas d'anomalies, les transformations géométriques ou les modifications de l'intensité sur l'image peuvent engendrer des défauts d'aliassage sur la nouvelle image.

II.4 Système visuel humain

Dans le domaine de l'échantillonnage et de la restitution, il faut faire une distinction entre un signal quelconque et une image et ce, surtout dans la phase de restitution. En effet, la restitution d'un signal est obtenue par l'interpolation de ses échantillons avec un filtre passe-bas idéal de restitution (cf. annexe A), tandis que dans un système à base de mémoire de trame (cf. figure II.3), l'interpolation des échantillons affichés sur le moniteur est faite à la fois par l'œil humain et par le spot lumineux non ponctuel du balayage (formé par l'excitation des luminophores de l'écran) ayant une distribution du type gaussienne. De plus, les diverses caractéristiques du système visuel humain influent directement sur la perception des résultats obtenus en synthèse d'images, en particulier pour les méthodes d'échantillonnage stochastique. Il convient donc d'insister sur certains aspects du système visuel de l'homme. C'est ce que nous allons faire maintenant, en nous inspirant de [CORN 71], de [GRAH 65] et de [YELL 83].

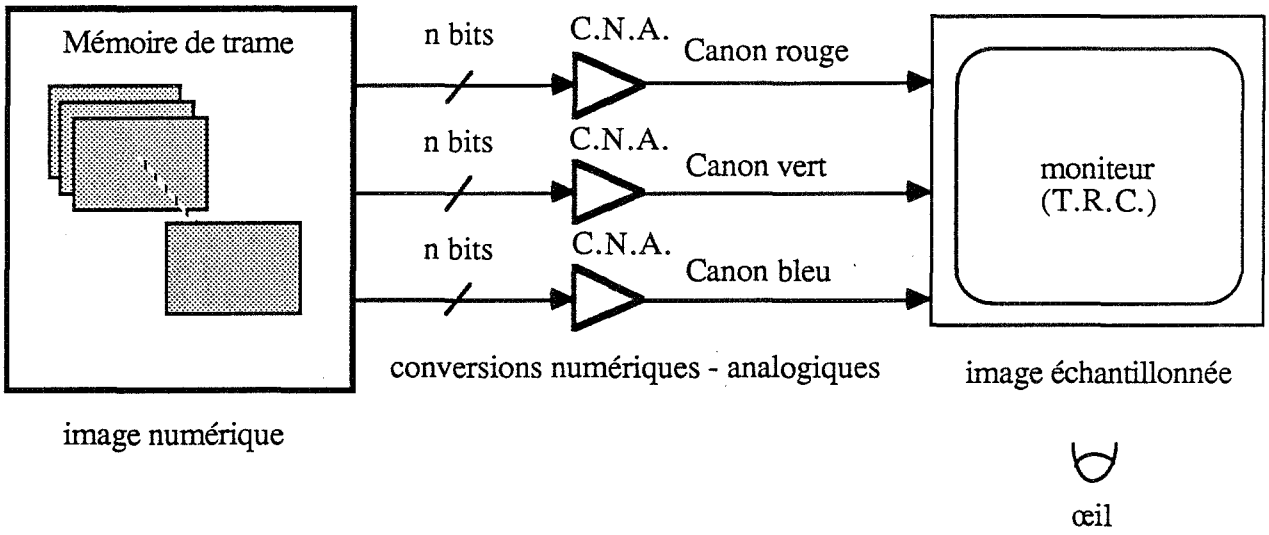


Figure II.3 : Restitution d'une image définie dans une mémoire de trame.

1- La réponse quasi-logarithmique :

La réponse du système œil-cerveau aux changements d'intensité n'est pas linéaire : elle est quasi logarithmique (cf. figure II.4). Par conséquent, l'œil est davantage sensible aux défauts d'aliassages (entre autres) dans les zones sombres. Ce phénomène peut être constaté en diminuant et en augmentant la luminosité du moniteur.

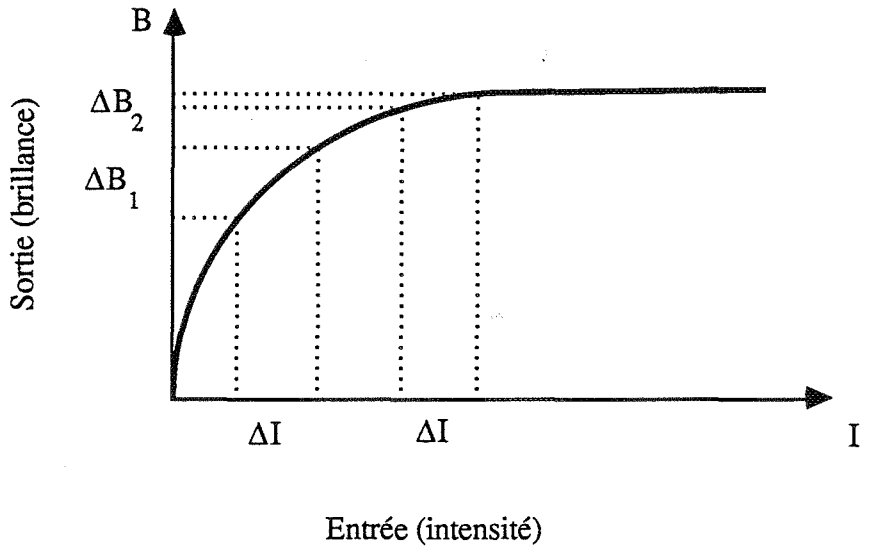


Figure II.4 : Réponse du système œil-cerveau.

2- Sensibilité aux zones de transition :

La sensibilité relative du système visuel aux fréquences spatiales est représentée par la figure II.5. A cause de cette caractéristique du système visuel, la distribution d'intensité décrite par la figure II.6.a sera finalement perçue comme celle que montre la figure II.6.c. En fait, la distribution présentée figure II.6.a peut être considérée comme ayant deux composantes d'intensité (cf. figure II.6.b). Le système visuel atténue les basses fréquences spatiales représentées par la composante en pointillés. Il y aura une deuxième atténuation sur les très hautes fréquences spatiales (celles qui correspondent aux minima de la composante "dents de scie"). Le résultat (cf. figure II.6.c) montre que les zones de transition d'intensité sont perçues avec une certaine accentuation (comme l'effet de Mach). Par conséquent, les défauts dus à l'aliassage comme le phénomène des marches d'escalier irrégulières sur la frontière entre deux zones d'intensités différentes (cf. chapitre III) sont très perceptibles pour le système œil-cerveau.

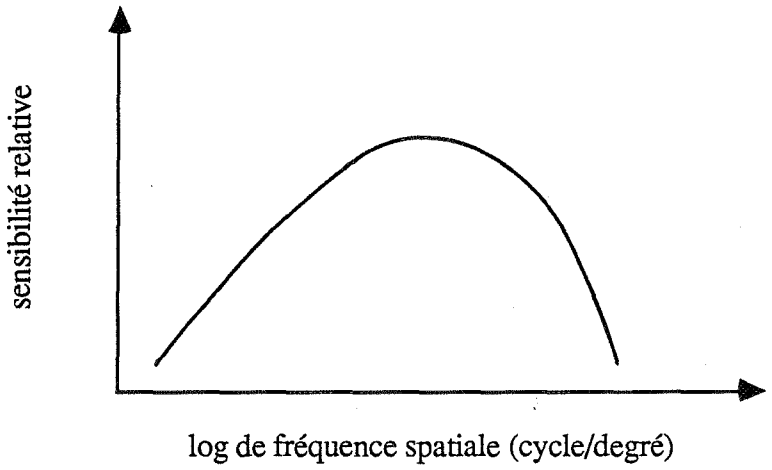


Figure II.5 : Sensibilité relative.

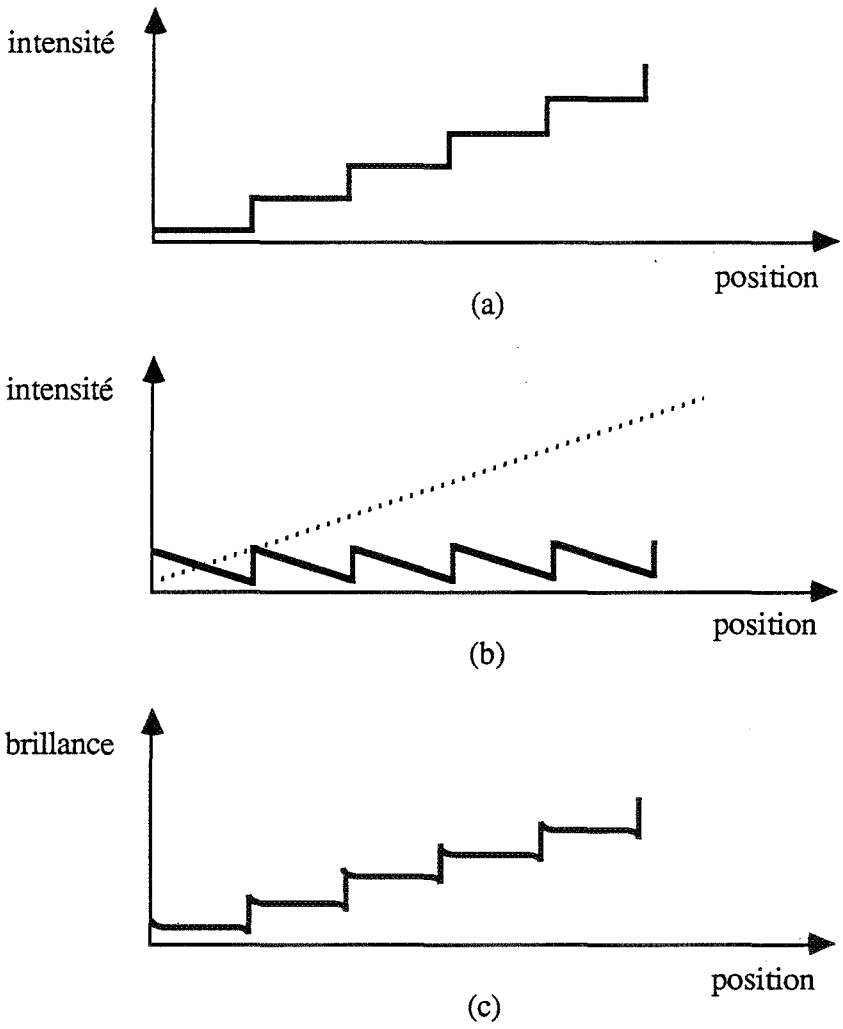


Figure II.6 : Accentuation des zones de transition (d'après CORNSWEET [CORN 71).

3- Distribution de Poisson :

L'œil humain dispose d'un certain nombre de photorécepteurs et par conséquent d'un nombre limité d'échantillons représentant une image. Pourtant, les images ne sont pas perçues comme aliassées par le système œil-cerveau humain. En fait, un filtrage passe-bas est effectué par le cristallin. De plus, la distribution spatiale des photorécepteurs est non uniforme. Grâce à cette distribution aléatoire appelée poissonnienne et malgré le nombre limité des photorécepteurs, l'aliassage peut être évité par un échantillonnage stochastique (cf. chapitre 1). La distribution de Poisson est à l'origine des méthodes d'antialiassage en synthèse d'images utilisant un échantillonnage stochastique.

II.5 Méthodes générales d'antialiassage en synthèse d'images

Nous allons étudier pour la synthèse d'images, dans un cadre général, chacune des trois solutions proposées dans le chapitre I pour résoudre les problèmes d'aliassage :

- l'augmentation des fréquences d'échantillonnage ;
- la limitation du spectre fréquentiel de la scène ;
- l'échantillonnage stochastique.

II.5.1 Augmentation des fréquences d'échantillonnage de la scène

Un pixel étant assimilé à un échantillon d'une image (un signal bidimensionnel), cette solution consiste à augmenter la définition de la mémoire de trame. On peut résumer cette technique de la manière suivante :

Scène R^2

↓ échantillonnage haute définition

Image numérique haute définition

↓

Affichage sur un support haute définition

Cette augmentation de la définition, outre le fait qu'elle est très coûteuse du point de vue algorithmique et matériel, pose de nombreux problèmes technologiques. Pour mettre en évidence ces contraintes technologiques qui imposent une faible définition aux mémoires de trame, on peut étudier l'affichage d'une image plutôt haute définition sur un moniteur du type télévision (tube à rayons cathodiques) utilisant un balayage de trame. Si la définition de l'image est 4000x4000 pixels et si la fréquence de balayage de l'image entière est de 50 Hz, la fréquence de la conversion numérique-analogique doit être au moins égale à 800 MHz ! D'ailleurs, une mémoire de trame d'accès très rapide est nécessaire. En fait, un moniteur du type télévision n'est pas adapté aux applications demandant une grande définition : en général le nombre de lignes est de l'ordre de 1000. Il faut donc utiliser d'autres types de supports (par exemple le film) pour l'affichage des images issues d'une mémoire de trame haute définition. L'utilisation d'un tel type de support pour l'affichage haute définition interdit les applications en temps réel et nous prive pratiquement d'interactivité, deux qualités souvent indispensables.

L'augmentation de la définition peut améliorer sensiblement certains défauts d'aliassage tels que les effets de marches d'escalier ou les problèmes d'affichage de petits objets (cf. chapitre III). Cependant, dans les cas tels que la présence de moirés (cf. chapitre V) où le phénomène d'aliassage est très important, ces défauts peuvent être toujours visibles, quelle que soit la définition de la mémoire de trame utilisée, le nombre d'échantillons n'étant pas encore suffisant. Bien que l'augmentation de la définition de la mémoire de trame améliore la qualité des résultats obtenus, la solution consistant à limiter le spectre avant l'échantillonnage de l'image demeure indispensable.

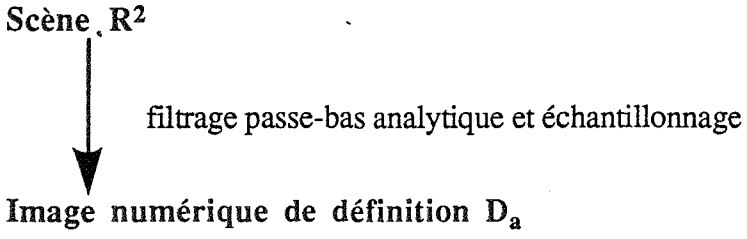
II.5.2 Limitation du spectre fréquentiel de la scène

La limitation du spectre fréquentiel de la scène (avant son échantillonnage) permet de sauvegarder une image dont le nombre d'échantillons est le nombre de pixels de la mémoire de trame. Pour limiter le spectre fréquentiel, on utilise un filtre passe-bas dont les fréquences de coupure sont égales à la fréquence spatiale maximale admissible. Si on dispose d'une mémoire de trame nxn, la fréquence maximale admissible sera donc :

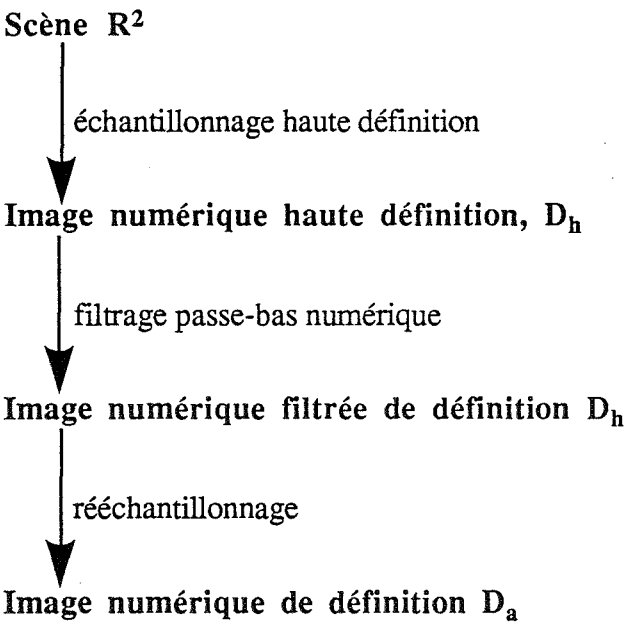
$$f_{max} = \frac{f_e}{2} = \frac{n}{2} \qquad \frac{\text{cycles}}{\text{largeur}} \tag{II.3}$$

où f_e est la fréquence d'échantillonnage. Pour une largeur de n (pixels), f_{max} est égale à 0.5 cycle par unité de distance. Par exemple, un signal sinusoïdal de la longueur d'onde (période) de deux pixels (selon chaque direction x et y) est représentable avec un échantillon (pixel) par l'alternance et un signal sinusoïdal d'une longueur d'onde inférieure ne l'est pas.

Le préfiltrage de la scène nous permettra de remplacer un échantillonnage ponctuel par un échantillonnage "surfacique" ("area sampling") de la scène. En fait, une fois la scène filtrée, chaque point contient un ensemble d'informations relatif à ce point et à ses voisins sur la scène originale. On peut donc dire qu'un échantillonnage ponctuel sur la scène filtrée est équivalent à un échantillonnage "surfacique" sur la scène originale (non-filtrée). Dans certains cas, on peut passer directement de la scène R^2 à l'image numérique de définition D_a (définition d'affichage), suivant le schéma suivant :



La scène originale est équivalente à une description numérique souvent haute définition (D_h) représentant "fidèlement" la scène sous sa forme analogique. Pour l'affichage de l'image, il faut finalement échantillonner la scène à la définition d'affichage (D_a) ; en général on choisit $D_h >> D_a$. Une deuxième solution est le passage par l'intermédiaire de l'image numérique D_h :



Nous allons maintenant étudier la solution du préfiltrage numérique dans le domaine fréquentiel puis dans le domaine spatial.

II.5.2.1 Préfiltrage numérique dans le domaine fréquentiel

Cette solution qui utilise le produit simple entre le spectre de l'image et la fonction de transfert du filtre (cf. chapitre I), nécessite le calcul de la transformée de Fourier discrète et bidimensionnelle de l'image numérique de définition D, puis celui de la transformée de Fourier inverse, une fois le filtrage effectué. Le calcul de chaque transformée de Fourier rapide ("FFT") nécessite $2D^2.\log_2 D$ opérations. Comme la valeur de D peut être très élevée (souvent $D=D_h>>D_a$), le préfiltrage dans le domaine fréquentiel demande un nombre d'opérations très important. Il est donc préférable d'effectuer le filtrage dans le domaine spatial.

II.5.2.2 Préfiltrage numérique dans le domaine spatial

Cette solution utilise le produit de convolution bidimensionnel discret (cf. chapitre I) d'une image numérique de définition D et de la réponse impulsionnelle d'un filtre passe-bas :

$$I_f(i,j) = h(i,j) ** I(i,j)$$

$$= \sum_{m=-\infty}^{m=+\infty} \sum_{n=-\infty}^{n=+\infty} h(m,n).I(i-m,j-n) \qquad \text{(II.4)}$$

où I est l'image numérique de définition D, I_f l'image filtrée et h la réponse impulsionnelle du filtre. Le filtre utilisé est un filtre passe-bas à réponse impulsionnelle finie [-M,M']x[-N,N'] (un tel filtre est noté RIF). La relation (II.4) peut s'écrire :

$$I_f(i,j)= \sum_{m=-M}^{m=+M'} \sum_{n=-N}^{n=+N'} h(m,n).I(i-m,j-n) \qquad \text{(II.5)}$$

De plus, on suppose que la fonction h est séparable, c'est à dire qu'elle vérifie :

$$h(m,n)= h_1(m).h_2(n) \qquad \text{(II.6)}$$

La fenêtre (spatiale) qui correspond au support du filtre à RIF s'appelle un masque. Le filtrage passe-bas par la relation (II.5) peut s'obtenir très simplement par superposition du masque et de l'image de telle sorte que le point (0,0) du masque coincide avec le point (i,j) de l'image. On utilise très souvent des masques impairs avec M=M' et N=N' pour lesquels le point (0,0) est le centre du masque (cf. figure II.7). Les valeurs h(m,n) sont généralement précalculées pour tous les points (m,n) du masque. Dans le cas où les temps de calcul sont importants, on peut même envisager de précalculer les valeurs h(m,n)xI(i,j) pour toutes les valeurs de m,n et I(i,j). Dans ce cas, la table de conversion aura trois entrées : m,n,I(i,j). L'opération de filtrage correspond à une moyenne pondérée (h est la fonction de pondération).

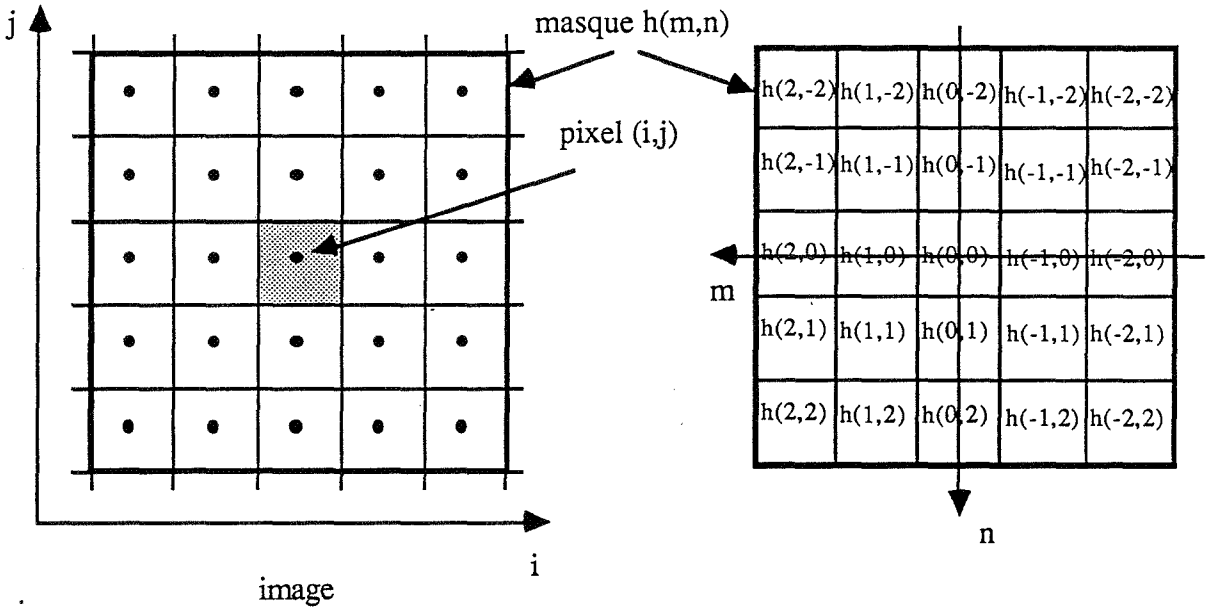


Figure II.7 : Superposition du masque et de l'image.

L'algorithme qui suit illustre le filtrage numérique d'images dans le domaine spatial. On peut remarquer que cet algorithme est général et qu'il peut être appliqué pour tout filtrage d'images numériques de définition quelconque.

Algorithme II.1, le filtrage d'une image numérique avec un masque (support) de taille impaire :

```
constantes
   max_col= nombre de colonnes-1;
   max_lig= nombre de lignes-1;
   marge= n     /* 2n+1= taille du masque */

type    octet : lig_buf[-marge..max_col+marge];

variables
      octets : ligne_image[0..max_col];
      entiers : x,y,i,j;
      réels : h[-marge..marge][-marge..marge];     /* le masque */
      lig_buf : buffer[-marge..marge];

{ /* opération de filtrage bidimensionnel */
  initialiser h;

  pour y:= -marge à max_lig+marge faire
  {
  si y<0 ou y>max_lig alors remise à zéro de buffer[marge]
  sinon affectation de la ligne y de l'image aux éléments 0 à max_col de buffer[marge];

  si y≥ marge alors
    { /* convolution et affichage */
      pour x:= 0 à max_col faire
       {
       ligne_image[x]:= 0;
       pour j:= -marge à marge faire
        pour i:= -marge à marge faire
          ligne_image[x]:= ligne_image[x]+round(buffer[-j][x-i].h[j][i]); /* on peut également
                                          utiliser une table de conversion afin d'économiser une multiplication*/
       }
      écriture de la ligne y-marge de l'image avec ligne_image;
    }

   pour j:= -marge à marge-1 faire
     buffer[j]:= buffer[j+1]; /* décalage vers le haut dans le buffer */
  }
}
```

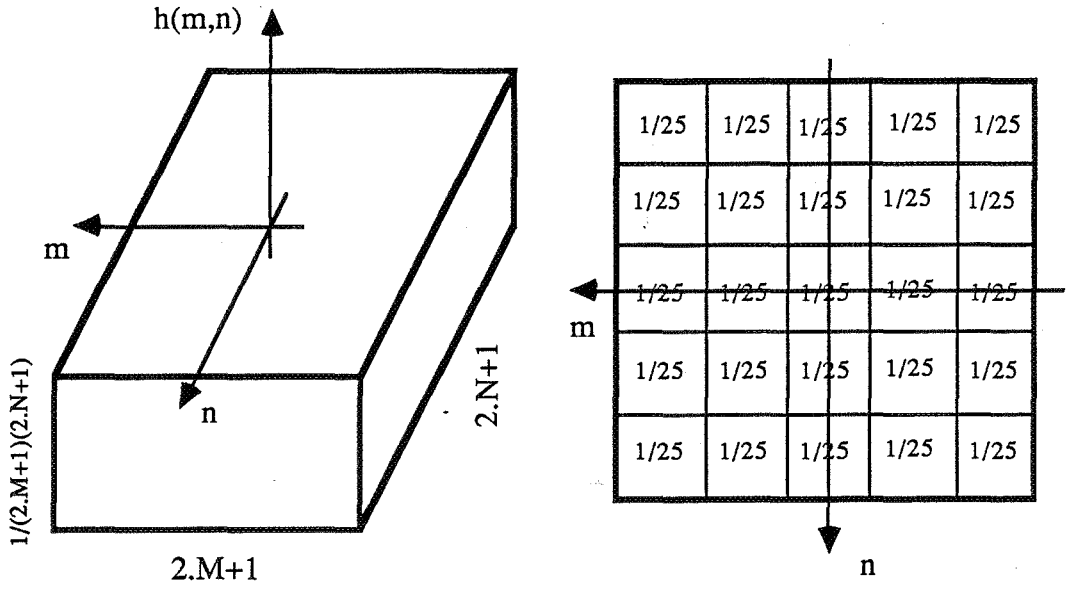
La fonction de pondération ("weighting function") et la taille du masque (support) sont les principaux paramètres du filtrage. Nous allons étudier les différents types de fonctions utilisées dans la pratique pour effectuer le filtrage passe-bas dans le domaine spatial. Pour de plus amples informations sur les filtres à RIF, on peut consulter [OPPE 75] et [RABI 75] .

1- Fonction de Fourier :

C'est la fonction la plus simple. La fenêtre normalisée de Fourier correspond à une fonction normalisée qui représente une boîte dont la base est le masque [-M,M]x[-N,N] la hauteur étant égale à $\frac{1}{[(2.M+1).(2.N+1)]}$ (cf. figure II.8.a) :

$$h(m,n) = \begin{cases} \frac{1}{[(2.M+1).(2.N+1)]} & \text{si } |m| \leq M \text{ et } |n| \leq N \\ 0 \quad \text{partout ailleurs} \end{cases} \qquad \text{(II.7)}$$

Le filtrage effectué à l'aide d'une fenêtre normalisée de Fourier est l'équivalent d'une moyenne simple ; cependant il est souvent utilisé pour des raisons de simplicité. L'exemple d'une fenêtre de Fourier pour un masque 5x5 est représenté par la figure II.8.b. On peut noter que l'utilisation d'une fenêtre de Fourier avec un masque de taille 2^m peut être intéressante car l'opération de filtrage ne nécessite alors que 2^m additions et m décalages à droite.



(a) représentation spatiale.

(b) représentation du masque et des valeurs de la fonction (RIF).

Figure II.8 : Fenêtre de Fourier normalisée.

2- Fonction de Bartlett :

C'est une fonction "pyramidale" définie par :

$$h(m,n)=h_1(m).h_2(n)$$

avec

$$h_1(k) = \begin{cases} 1-\frac{|k|}{(K)} & \text{si } |k|\leq K \\ 0 & \text{si } |k|>K \end{cases} \qquad \text{(II.8)}$$

$h_1(k)$ est représentée par la figure II.9. On dit que la fenêtre de Bartlett est normalisée si le volume de la pyramide définie par la fonction est égal à un. Le filtrage avec une telle fenêtre est équivalent à une moyenne pondérée.

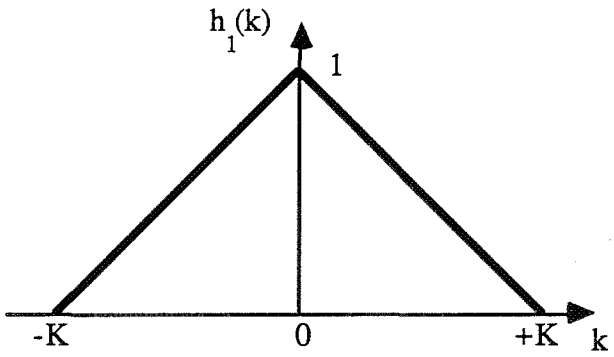


Figure II.9 : Fenêtre de Bartlett monodimensionnelle.

Pour un exemple d'utilisation de fenêtres de Bartlett ayant des définitions différentes, on peut consulter les images présentées dans [CROW 81].

3- Fonction de Hamming :

La fenêtre généralisée de Hamming [WEBS 78] est décrite par la relation suivante :

$$h(m,n)= h_1(m).h_2(n)$$

avec

$$h_1(k) = \begin{cases} \alpha+\beta.\cos^p(\frac{\pi.k}{K}) & \text{si } |k|\leq K \\ 0 & \text{si } |k|>K \end{cases} \qquad \text{(II.9)}$$

où $\alpha+\beta = 1$ et p est un constante. La fenêtre de Hamming est généralement définie par $\alpha = 0.54$, $\beta = 0.46$ et $p = 1$:

$$h_1(k) = \begin{cases} 0.54+0.46\cos(\frac{\pi.k}{K}) & \text{si } |k| \leq K \\ 0 & \text{si } |k|>K \end{cases} \qquad \text{(II.10)}$$

La figure II.10 montre $h_1(k)$. L'utilisation d'une fenêtre normalisée de Hamming équivaut à une moyenne pondérée.

L'image III.3 est le résultat d'un filtrage à l'aide d'une fenêtre de Hamming.

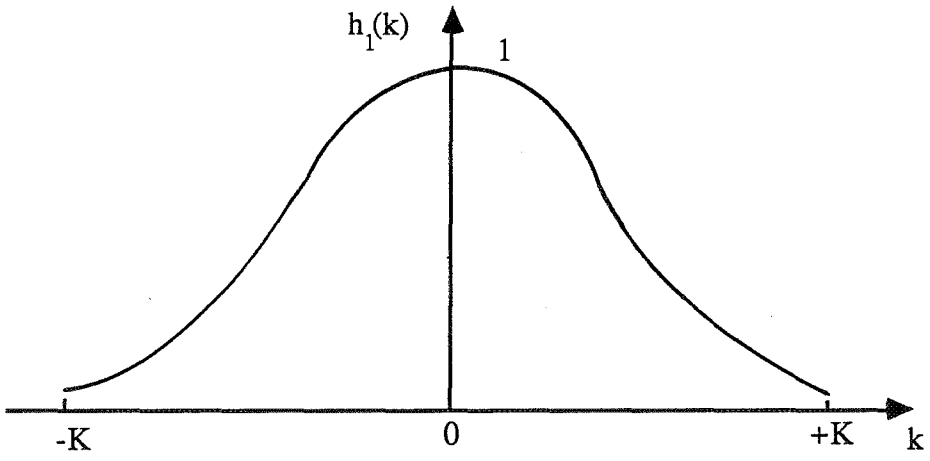


Figure II.10 : Fenêtre de Hamming (α= 0.54 et β= 0.46).

Il existe d'autres fonctions utilisées, dont les plus connues sont la fonction conique (cf. [FEIB 80]), la fonction bidimensionnelle Sinc (cf. chapitre I), les fonctions cubiques (cf. [MITC 88]) et la fonction bidimensionnelle gaussienne (cf. [KAJI 81]). En pratique, il n'existe pas de règle absolue pour le choix d'une fenêtre parmi celles qui sont connues [BLIN 89]. En fait, ce choix dépend de nombreux facteurs objectifs et subjectifs comme le support d'affichage, la structure de l'image, la sauvegarde des hautes fréquences ou le choix de couleurs et de textures. Cependant, le filtrage obtenu avec la fenêtre de Fourier est assez grossier et on essaie plutôt d'effectuer un filtrage utilisant des fonctions plus sophistiquées.

II.5.3 Echantillonnage stochastique

Les méthodes classiques d'antialiassage peuvent être très coûteuses en temps de calcul pour certaines techniques de visualisation comme le lancer de rayons (cf. chapitre III).

L'échantillonnage stochastique (cf. chapitre I) est une alternative qui peut diminuer sensiblement les défauts dûs à l'aliassage avec des temps de calcul plutôt raisonnables [DIPP 85] [COOK 86] [MITC 87].

L'implémentation la plus simple, mais la plus coûteuse, d'un échantillonnage stochastique consiste à générer et à stocker dans un tableau une grille d'échantillonnage aléatoire en utilisant la distribution de Poisson et en respectant une restriction sur la distance minimale : tout nouveau point d'échantillonnage situé à une distance inférieure à la distance minimale d'un point déjà existant sera écarté de la grille [COOK 86]. Un filtre passe-bas normalisé détermine l'influence d'un échantillon sur les pixels voisins. Les valeurs de ce filtre sont calculées et stockées dans un tableau. Cette méthode nécessite donc un espace mémoire considérable, surtout avec une grille d'échantillonnage d'une définition généralement supérieure à celle d'affichage. La distribution de Poisson peut être remplacée par l'ajout de bruit sur une grille d'échantillonnage régulière ("jittering") afin d'obtenir une grille perturbée [DIPP 85] [COOK 86]. Cette technique, dans le cas monodimensionnel temporel, est étudiée dans [BALA 62] et [COOK 86]. Si la période d'échantillonnage est T_e, le n-ième échantillon est pris à l'instant $n.T_e+p_n$ au lieu de T_n. Les valeurs de perturbation p_n doivent être non-corrélées.

Les études faites dans [BALA 62] ont montré les propriétés suivantes pour l'échantillonnage avec perturbation d'une grille régulière :

- les composantes hautes fréquences du spectre sont atténuées ;
- l'énergie des parties atténuées apparaît sous forme de bruit blanc ;
- l'allure de base du spectre reste inchangée.

De plus, [BALA 62] signale que l'échantillonnage par perturbation d'une grille régulière ne peut pas être considéré comme un filtrage, car il n'est pas linéairement invariant par translation. Cependant, la perturbation de la grille d'échantillonnage suivi d'un filtrage idéal de restitution (cf. chapitre I) est un processus linéairement invariant par la translation. L'atténuation des composantes hautes fréquences peut donc être considérée dans une telle situation. Par conséquent, tout échantillonnage stochastique est suivi par un filtrage passe-bas.

Deux types de fonctions non-corrélées sont étudiées : la perturbation gaussienne et le bruit blanc [BALA 62] [COOK 86]. Pour la perturbation par la fonction gaussienne (cf. figure II.11.a), l'atténuation des composantes fréquentielles s'obtient par la relation suivante :

$$A(f)=\frac{1}{\exp(2\pi f\sigma)^2} \qquad \text{(II.11)}$$

où σ^2 est la variance de la distribution gaussienne.

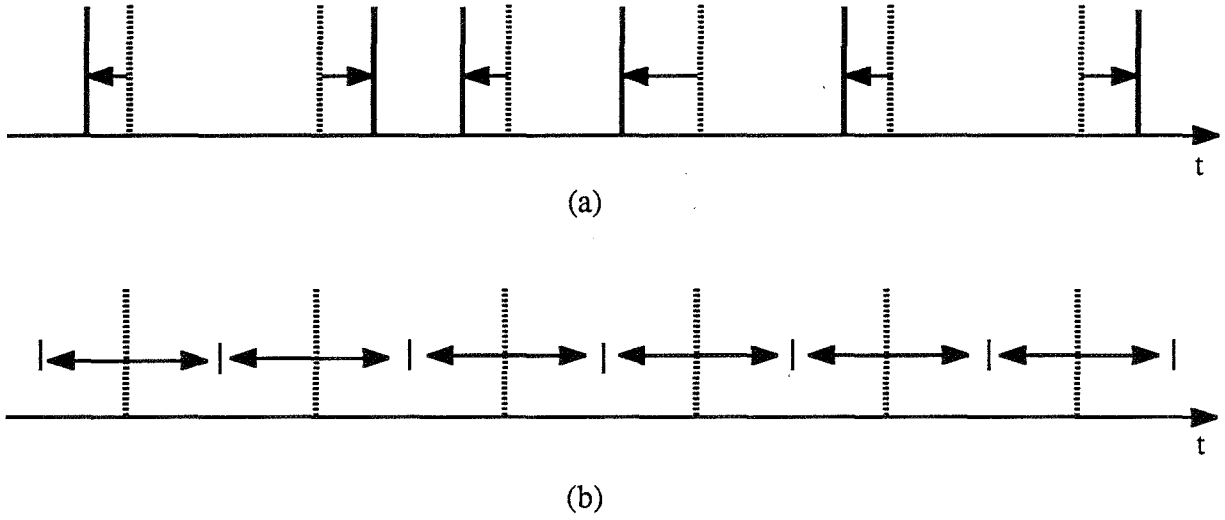


Figure II.11 : Fonctions de perturbation.

La figure II.12.a présente la fonction d'atténuation pour la valeur $\sigma=\frac{T_e}{6,5}$.

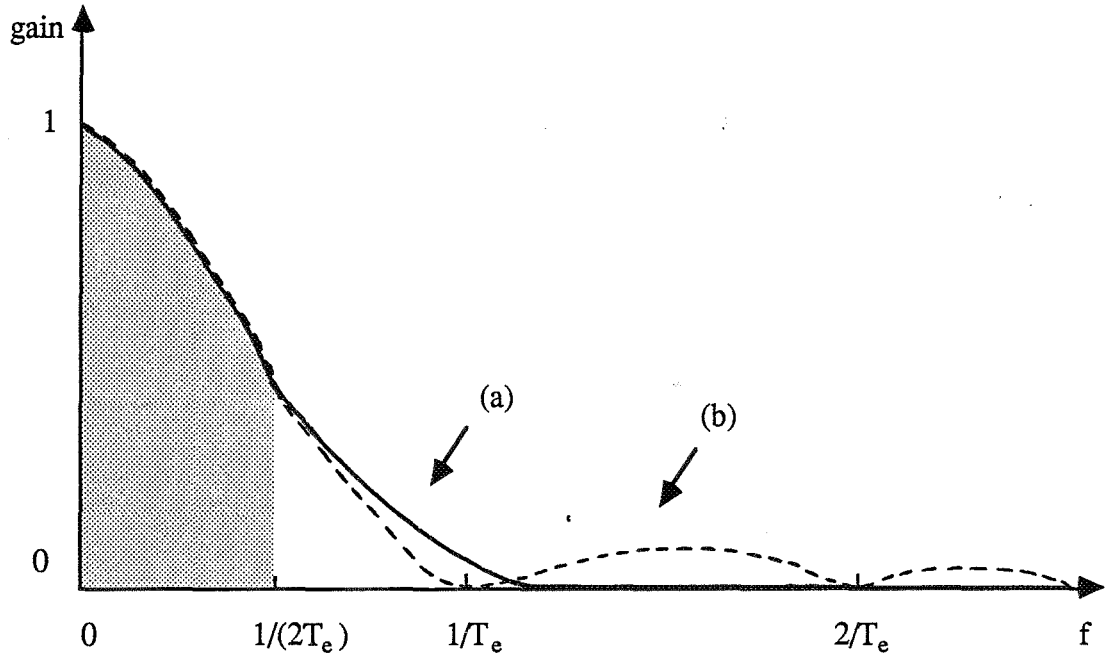


Figure II.12 : Fonctions d'atténuation.

Pour la perturbation par le bruit blanc, les valeurs de P_n sont distribuées uniquement entre $-\gamma T_e$ et γT_e (cf. figure II.11.b). Dans ce cas, l'atténuation des composantes fréquentielles est donnée par la relation suivante :

$$A(f)= Sinc^2(2\pi f\gamma T_e) \qquad (II.12)$$

La figure II.12.b présente cette atténuation pour la valeur γ=0,5.

La figure II.12 montre que la perturbation d'une grille régulière ne produit pas un antialiassage total mais réduit l'aliassage. Le changement des valeurs γ ou σ permet de modifier la largeur du filtre afin d'avoir un compromis entre l'aliassage et le bruit (cf. chapitre 1).

L'échantillonnage par perturbation d'une grille régulière peut être facilement généralisé au cas bidimensionnel. Considérons une grille d'affichage régulière des pixels carrés (cf. figure II.1) ; chaque pixel de cette grille peut être également considéré comme une grille régulière d'un ou plusieurs sous-pixels carrés. Le centre de chaque sous-pixel (x,y) est un point d'échantillonnage régulier. Le bruit peut être ajouté d'une façon indépendante à x et à y afin de perturber la grille d'échantillonnage. Chaque point d'échantillonnage $(x+p_x,y+p_y)$ de la nouvelle grille est donc situé aléatoirement à l'intérieur d'un sous-pixel (x,y) et sa couleur sera déterminée en fonction de la couleur de l'entité de la scène visible au point $(x+p_x,y+p_y)$.

Pour trouver les couleurs des pixels de la grille d'affichage, les valeurs d'échantillonnage déterminées sur la grille aléatoire seront ensuite filtrées par un filtre passe-bas normalisé. Pour des raisons de simplicité, des filtres passe-bas précalculés avec un support (masque) régulier (cf. paragraphe II.5.2.2) peuvent être utilisés. En fait, si l'étendue du support du filtre est important comparé à la perturbation de la grille d'échantillonnage, l'effet de cette perturbation sur le filtre peut être négligé.

Une autre méthode pour approximer une distribution poissonienne est donnée dans [MITC 87]. Elle s'inspire d'une méthode déjà utilisée pour la simulation des niveaux de gris basée sur une distribution d'erreur aux pixels voisins d'un pixel donné [FLOY 75]. Pour trouver une grille irrégulière avec un échantillonnage par pixel d'affichage, cet algorithme utilise une grille régulière 4x4 fois plus fine que celle d'affichage. Les points d'échantillonnage sont sélectionnés sur cette grille. L'algorithme procède par ligne de balayage. A chaque point de la grille fine est associé une valeur de diffusion $D_{i,j}$ qui est calculée à partir des autres valeurs calculées auparavant et d'une source de bruit R :

$D_{i,j}$ = T-SELECT

où

$$T = \frac{4.D_{i-1,i} + D_{i-1,i-1} + 2.D_{i,i-1} + D_{i+1,i-1}}{8} + R$$

et

$$SELECT = \begin{cases} 0 \text{ si } T < 0,5 \\ 1 \text{ sinon} \end{cases}$$

Si SELECT=1, le point de la grille fine est un point d'échantillonnage. On constate qu'à chaque étape, le stockage de deux lignes successives de balayage est nécessaire.

Signalons finalement que la grille d'échantillonnage stochastique a une définition qui est souvent supérieure à celle de la grille d'affichage.

II.6 Conclusion

Les problèmes d'aliassage en synthèse d'images sont dus à un échantillonnage inapproprié (insuffisant). L'augmentation de la définition de la mémoire de trame est limitée par les contraintes technologiques. En plus, elle ne permet pas de résoudre le phénomène d'aliassage dans certains cas tels que les moirés. La solution retenue est donc le filtrage passe-bas de la scène avant son échantillonnage afin de limiter le spectre fréquentiel de l'image. Etant donné que le filtrage dans le domaine fréquentiel est coûteux, on utilise le filtrage dans le domaine spatial (comme c'est le cas pour les images de télévision). Ce type de filtrage peut être effectué avec des filtres passe-bas bidimensionnels à réponse impulsionnelle finie (RIF). Dans certains cas, l'échantillonnage stochastique, basé sur la distribution de Poisson, est une autre approche de l'antialiassage qui permet de diminuer sensiblement les défauts visibles par un observateur. La distribution de Poisson peut être remplacée par la perturbation d'une grille d'échantillonnage régulière, suivie par un filtrage de restitution.

Références du chapitre II

[BALA 62] : A. BALAKRISHNAN,
"On the Problem of Time Jitter in Sampling", IRE Transacations on Information Theory, April 1962, pp. 226-236.

[BLIN 89] : J. BLINN,
"Return of the Jaggy", IEEE Computer Graphics and Applications, vol. 9, No. 2, March 1989, pp. 82,89.

[CONR 80] : CONRAC DIVISION,
"Raster Graphics Handbook", Conrac Division, Conrac Corporation, 1980, chap. 9 and 10.

[CORN 71] : T.N. CORNSWEET,
Visual Perception, Academic Press, 1971, pp. 268-383.

[COOK 86] : R. COOK,
"Stochastic Sampling in Computer Graphics", ACM Transactions on Graphics, vol. 5, No. 1, January 1986, pp. 51-72.

[CROW 81] : F.C. CROW,
"A Comparison of Antialiasing Techniques", IEEE Computer Graphics and Applications, vol. 1, No. 1, January 1981, p. 40-48.

[DIPP 85] : M. DIPPE and E.WOLD,
"Antialiasing Trough Stochastic Sampling", Computer Graphics, vol. 19, No. 3, jully 1985, pp. 69-78.

[FEIB 80] : E. FEIBUSH, M.LEVOY and R. COOK,
"Synthetic Texturing Using Digital Filters", Computer Graphics, vol. 14, No. 3, July 1980, pp. 295-301.

[FLOY 75] : R. FLOYD and L. STEINBERG,
"An Adaptive Algorithm for Spatial Grey Scale", SID Digest, 1975, pp. 36-37.

[GRAH 65] : C. GRAHAM, N. BARTLETT, J. BROWN, Y.HSIA, C. MUELLER and L. RIGGS,
Vision and visual perception, John Wiley & Sons, Inc., 1965, pp. 321-345.

[KAJI 81] : J. KAJIA and M. ULLNER,
"Filtering High Quality Text for Display on Raster Scan Devices", Computer Graphics, vol. 15, No. 3, August 1981, pp. 7-15.

[MITC 87] : D. MITCHELL,
"Generating Antialiased Images at Low Sampling Densities", Computer Graphics, vol. 21, No. 4, July 1987, pp. 65-72.

[MITC 88] : D. MITCHELL and A. NETRAVALI,
"Reconstruction Filters in Computer Graphics", Computer Graphics, vol. 22, No. 4, august 1988, pp. 221-228.

[OPPE 75] : A. OPPENHEIM and R. SCHAFER,
Digital Signal Processing, Prentice-Hall, 1975, pp. 237-250.

[RABI 75] : L. RABINER and B. GOLD,
Theory and Application of Digital Signal Processing, Prentice-Hall, 1975, pp. 75-183.

[WEBS 78] : P.J. WEBSTER,
"A Generalized Hamming Window", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-26, No. 2, April 1978, pp. 176-177.

[YELL 83] : J. YELLOTT,
"Spectral Consequences of Photorecepteur Sampling in the Rhesus Retina", Science, vol. 221, July 1983, pp. 382-385.

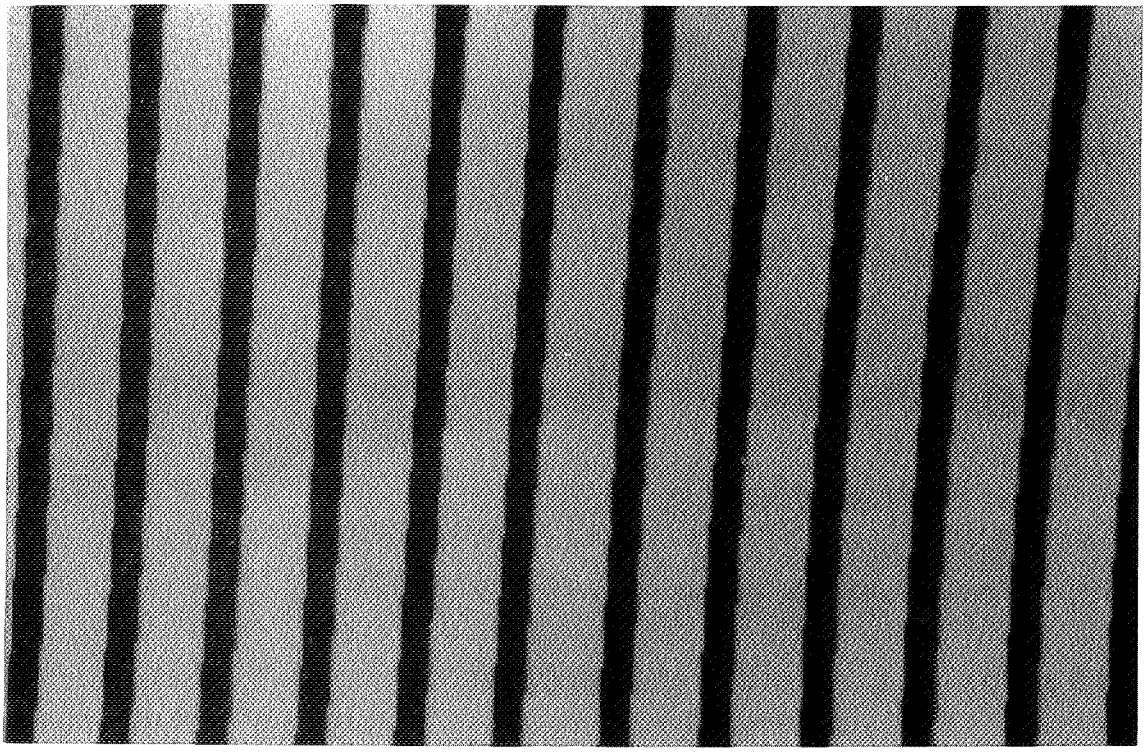


Image II.1 : Saisie binaire par caméra représentant l'effet d'aliassage.

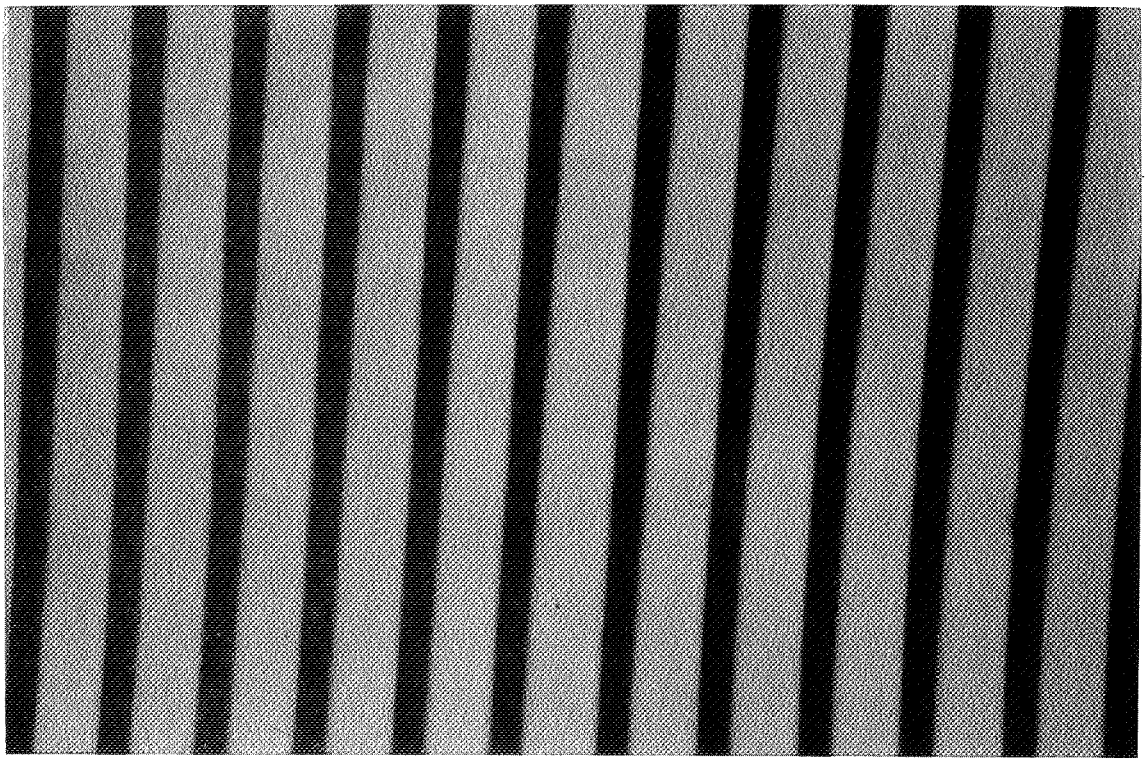


Image II.2 : Saisie par caméra en 256 niveaux d'intensité sans effet d'aliassage.

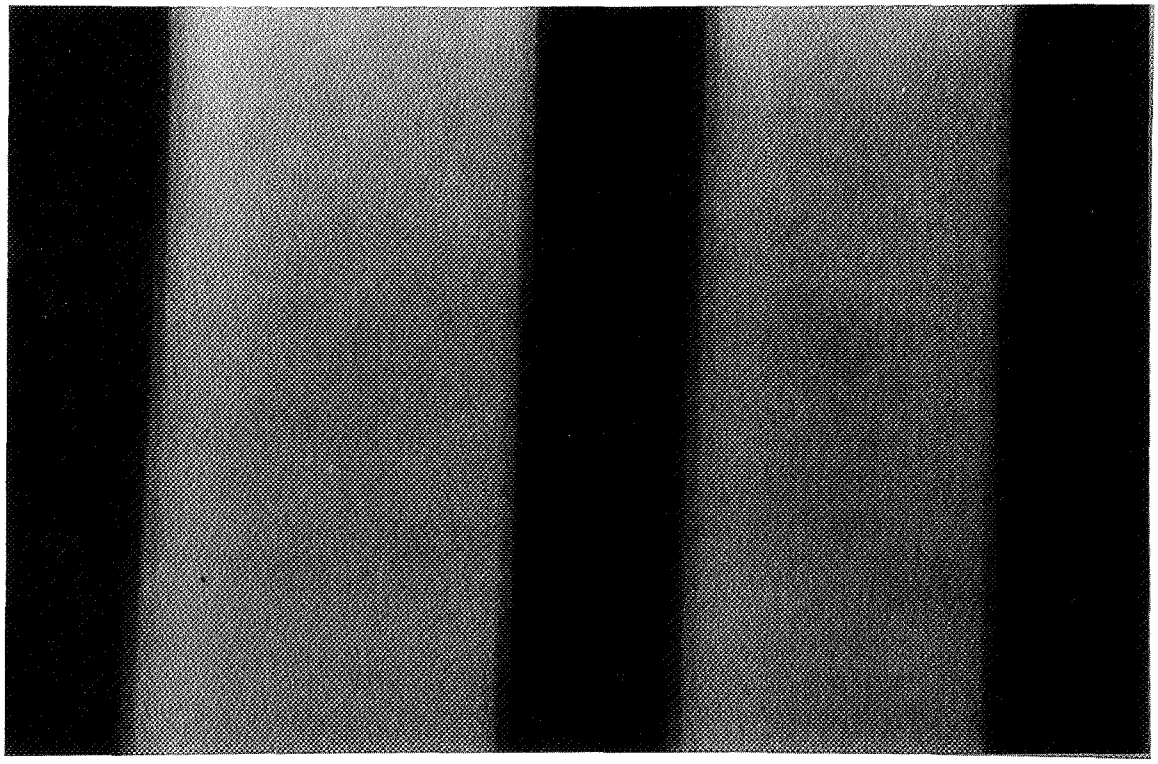


Image II.3 : Zoom x5 de l'image II.2.

CHAPITRE III

Aliassage et antialiassage d'objets bi ou tridimensionnels

CHAPITRE III

Aliassage et antialiassage d'objets bi ou tridimensionnels

III.1 Introduction

Dans le premier chapitre, nous avons étudié les problèmes de discrétisation des signaux bidimensionnels afin de trouver l'origine du phénomène d'aliassage : un échantillonnage insuffisant. Lors du chapitre précédent, nous avons abordé ce phénomène en synthèse d'images et nous avons montré qu'il est étroitement lié à la définition insuffisante des mémoires de trame imposée par des contraintes technologiques. Chaque changement abrupt de couleur (ou intensité) dans la scène correspond à une zone hautes fréquences de la scène. L'affichage d'une scène contenant des hautes fréquences spatiales dépassant les limites autorisées par cette définition provoque inévitablement de l'aliassage.

Deux des trois cas d'aliassage apparaissant en synthèse d'images seront abordés dans ce chapitre : les frontières entre deux zones de couleurs différentes et la présence de petits objets. Le cas de moiré pour des entités texturées sera étudié ultérieurement dans le chapitre V. Nous présenterons les différentes méthodes d'antialiassage dans un contexte 2D, sans nous préoccuper des problèmes liés à la méthode d'affichage ou d'élimination des parties cachées dans un premier temps. Ensuite, nous verrons l'application des méthodes d'antialiassage aux divers algorithmes de visualisation et nous proposerons les techniques d'antialiassage adaptées à chaque cas.

Rappelons qu'une scène est souvent facettisée avant son affichage à l'aide des algorithmes tels que celui de [LANE 80]. Nous étudierons donc plus particulièrement le cas des segments de droite et des contours polygonaux, qui sont les deux entités géométriques les plus couramment employées pour décrire les images à afficher.

Les différentes méthodes étudiées dans ce chapitre sont décrites pour des images en niveaux de gris ; pour des images en couleur, il est nécessaire d'appliquer la technique présentée à chacune des trois primitives rouge, verte et bleue de l'image.

III.2 Présentation des problèmes

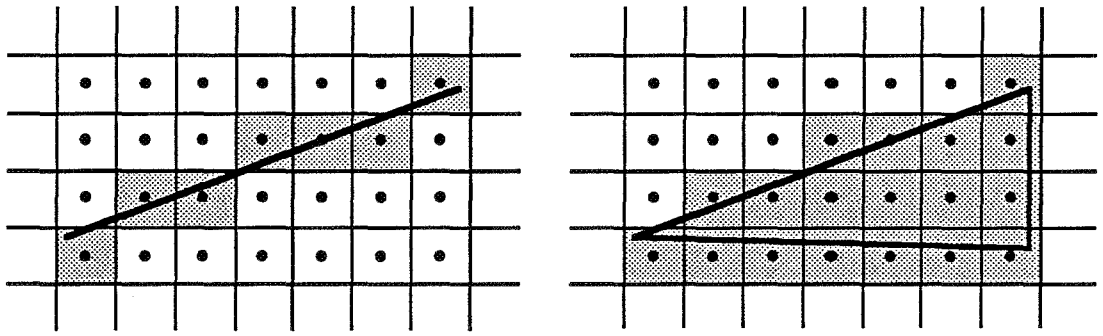
III.2.1 L'effet de marches d'escalier

Un des problèmes d'aliassage le plus couramment rencontré sur une image numérique est l'effet de marches d'escalier irrégulières ("jaggies") visibles sur la frontière entre deux zones d'intensité (ou couleur) différente. Nous allons décrire cet effet pour l'affichage d'une zone d'intensité $I_z(x,y)$ sur un fond d'intensité $I_f(x,y)$. L'algorithme d'affichage fournit l'ensemble des pixels qui représentent "au mieux" la zone sous sa forme discrète. Ces pixels sont ensuite affichés avec l'intensité $I_z(x,y)$. Ainsi, le caractère "tout ou rien" de l'algorithme d'affichage crée une frontière crénelée. L'ampleur du défaut dépend de la définition de la grille d'affichage. L'effet de marches d'escalier est plus important si la frontière entre deux zones représente un changement d'intensité considérable correspondant à des hautes fréquences de la scène. L'orientation de la frontière entre les deux zones peut aussi jouer un rôle important. En effet, le problème est présent pour toute orientation de la frontière à l'exception des orientations horizontale et verticale pour lesquelles les segments sont confondus avec une ligne ou une colonne de la grille de pixels. Les défauts sont très visibles pour des orientations presque parallèles aux axes alors que pour des orientations voisines de 45 degrés pour lesquelles la longueur des marches est plus petite (et donc mieux intégrée par l'œil) le phénomène est moins perceptible. Les images III.1 et III.2 montrent l'effet de marches d'escalier irrégulières pour deux scènes simples affichées sans aucune technique d'antialiassage, dans le cas général où le fond et la scène peuvent être texturés.

Des études montrent que l'œil humain est très sensible aux contrastes (cf. chapitre II). Par conséquent, même si la proportion de l'image occupée par des frontières est très faible, l'œil sera très sensible aux défauts de marches d'escalier présents dans ces zones (cf. images III.1 et III.2).

Le problème de marches d'escalier est général et apparaît presque systématiquement lors de l'affichage d'une frontière entre deux zones quelconques d'intensités différentes. Cependant, nous envisagerons plus particulièrement des entités simples telles que des segments ou des polygones. En effet, ces entités sont les plus courantes en synthèse d'image car elles peuvent être obtenues par décomposition d'éléments de degrés supérieurs, courbes ou surfaces. De plus, elles sont généralement plus faciles à traiter par les algorithmes

d'élimination des parties cachées et d'antialiassage. L'effet de marches d'escalier créé par un algorithme d'affichage classique, "tout ou rien", est le même pour des segments de droite que pour des bords de polygones car les bords d'un polygone sont déterminés par le même algorithme de tracé de segment lancé entre deux sommets successifs du polygone, (cf. figure III.1). Les résultats d'affichage par un tel algorithme [BRES 65] dans les deux cas sont représentés par les images III.1 et III.2.



(a) affichage d'un segment (b) affichage d'un polygone

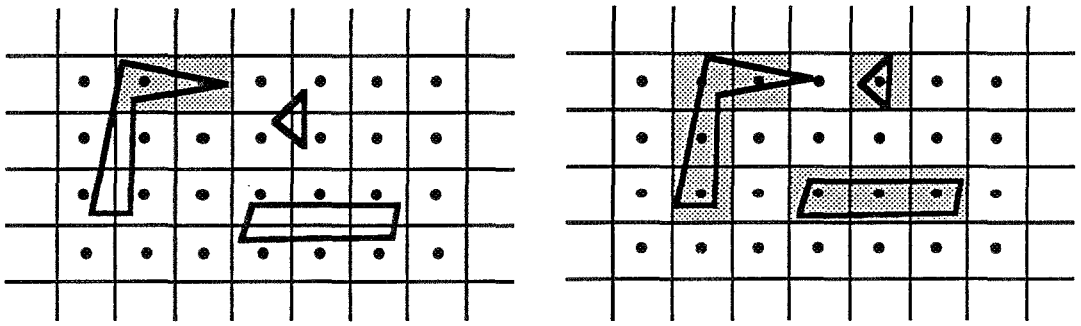
Figure III.1 : Affichage à l'aide d'un algorithme traditionnel "tout ou rien".

Afin d'éviter, ou de diminuer sensiblement, les problèmes de marches d'escalier, il est nécessaire de mettre en place un processus d'échantillonnage correct en utilisant généralement un filtre passe-bas antirepliement avant d'échantillonner la scène à la définition d'affichage (cf. chapitres précédents). En ce qui concerne le traitement d'antialiassage des segments et des bords de polygones, on peut appliquer un seul algorithme d'antialiassage qui traite à la fois les bords de polygones et les segments : un segment physique a une épaisseur finie, il peut donc être considéré comme un polygone avec un rapport $\frac{\text{longueur}}{\text{largeur}} >> 1$. Mais étant donnée l'importance des segments dans la définition des scènes, ils sont très souvent traités comme un cas particulier. Le résultat de toute technique appropriée d'antialiassage consiste à afficher les pixels proches de la frontière avec des intensités dégradées entre celle de la zone et celle du fond (cf. image III.6). Cette représentation des intensités intermédiaires est définie par le filtre passe-bas antirepliement (à RIF) utilisé (cf. chapitre II).

III.2.2 Petits objets

Un petit objet est un élément de la scène définie en R^2 ou R^3 dont l'épaisseur maximale après projection sur l'écran est localement inférieure à un pixel d'affichage. Les problèmes de discrétisation liés à la présentation de petits objets de la scène sont les suivants :

1- L'affichage d'un petit objet dépend de sa position dans la scène. En effet, les régions de l'objet qui n'intersectent pas de centre de pixels ne seront pas visibles dans l'image affichée (cf. figure III.2). Ce problème est important pour des images animées car, dans ce cas, les petits objets risquent d'apparaître, de disparaître ("flash"), de changer de forme ou de se décaler (saut de ligne ou de colonne) en fonction du temps, selon leur position dans la scène par rapport à la grille d'affichage.



(a) (b)

(a) petits objets invisibles (perdus entre les centres des pixels) ;
(b) petits objets affichés.

Figure III.2 : Affichage de petits objets selon leur position dans la scène.

2- L'apparition et la disparition de certaines régions de l'objet peut conduire à un affichage en pointillé des petits objets longs qui ont une épaisseur voisine d'un pixel (cf. figure III.3). Ce problème est aussi important pour des images dynamiques que pour des images statiques. Pour l'affichage de segments, ce problème est résolu en utilisant un algorithme de tracé de segment qui affichera toujours au moins un pixel par ligne ou par colonne selon la pente du segment.

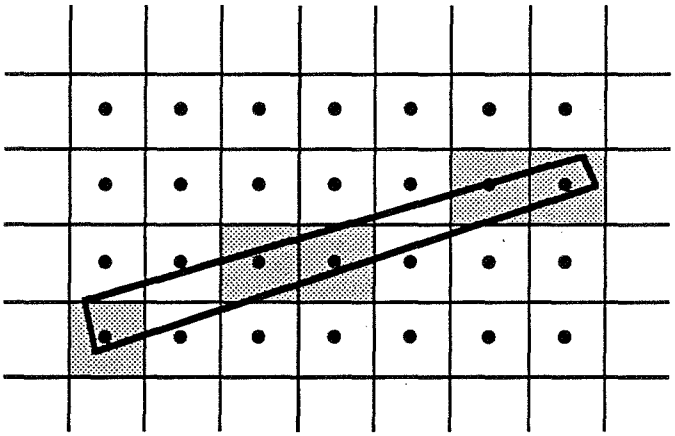


Figure III.3 : Un petit objet long peut apparaître en pointillé.

III.3 Méthodes d'antialiassage 2D

Nous allons étudier les techniques d'antialiassage 2D dans ce paragraphe. Ensuite, nous verrons les problèmes que posent l'application des techniques d'antialiassage dans le cas de certains algorithmes d'affichage.

III.3.1 Classification des méthodes de traitement

On peut distinguer trois classes de méthodes de traitement : le post-filtrage, le préfiltrage et l'échantillonnage stochastique.

Post-filtrage :

Dans ce cas, le filtrage passe-bas est appliqué sur l'image affichée avec la définition de la grille d'affichage (qui est donc une image déjà aliassée). Malgré sa simplicité, dans la majorité des cas, cette méthode n'offre pas une qualité acceptable, conformément à ce que laisse prévoir la théorie présentée dans le premier chapitre.

Préfiltrage :

Le préfiltrage passe-bas de la scène est la solution classique d'antialiassage. On distingue plus particulièrement deux classes de solutions:

- La première consiste à sur-échantillonner l'image puis à appliquer un filtrage. Cette méthode est particulièrement bien adaptée à un affichage par ligne de balayage.

- La deuxième solution est plutôt propre à des polygones ou des segments. Dans ce cas, les entités linéaires (bords de polygones et segments) sont affichées à l'aide d'un algorithme réalisant directement l'antialiassage.

Echantillonnage stochastique :

Cette technique peut être utilisée indifféremment pour les cas bi ou tridimensionnel mais étant donnés les bons rapports qualité/temps offerts par les méthodes de préfiltrage 2D, celle-ci est plutôt employée dans le cas 3D, en particulier avec le lancer de rayons (cf. paragraphe III.4.4). Nous verrons donc cette technique ultérieurement dans le paragraphe III.4.4.

III.3.2 Post-filtrage

Lors de l'affichage d'une image effectué sans traitement particulier, la frontière entre deux objets d'intensités différentes est généralement crénelée. Ce problème, qui est dû à un changement d'intensité important, peut être rendu moins visible par un changement plus doux, en plusieurs étapes. L'application d'un post-filtrage passe-bas sur une image crénelée répond à ce besoin. En effet, il permet d'adoucir les frontières. Ce résultat peut être interprété en reprenant les discussions théoriques du premier chapitre. Pour des raisons de simplicité, considérons le cas d'un signal monodimensionnel (cf. figure III.4). La zone hachurée entre les fréquences f_0 et f_{max} représente le repliement du spectre du signal, c'est à dire la zone aliassée. Un filtrage passe-bas de fréquence de coupure $f_c = \frac{f_e}{2}$ permet d'éviter l'aliassage s'il est appliqué avant l'échantillonnage du signal ; par contre, il devient beaucoup moins efficace s'il est appliqué sur le signal aliassé, sans conserver davantage de hautes fréquences. Pour mieux atténuer le phénomène d'aliassage, il est nécessaire d'utiliser un filtre passe-bas dont la bande passante est plus étroite, de fréquence de coupure située entre f_0 et $\frac{f_e}{2}$.

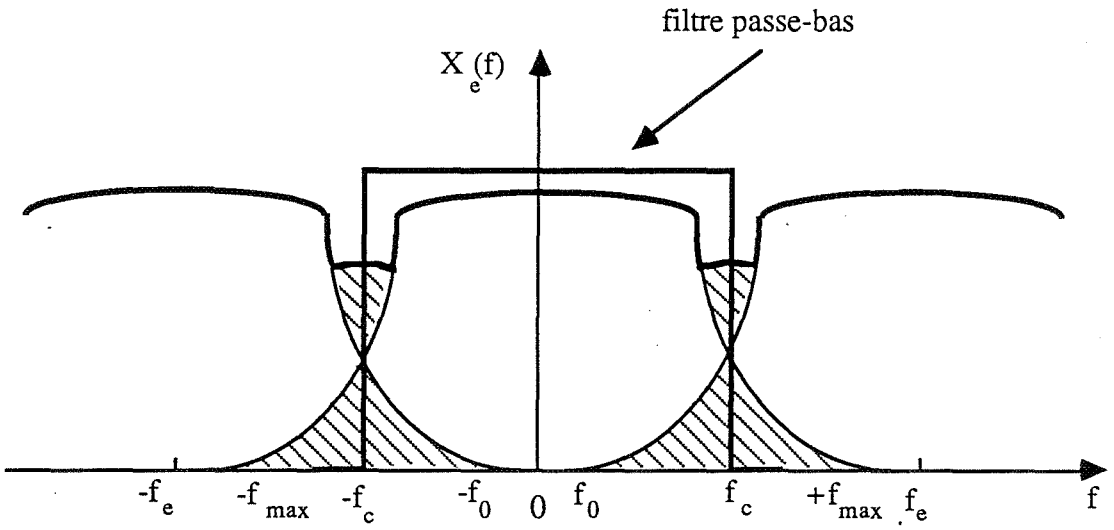


Figure III.4 : Aliassage et post-filtrage.

Cependant, dans le cas où l'aliassage (repliement du spectre) n'existe pas, l'application d'un post filtrage passe-bas de fréquence de coupure $\frac{f_e}{2}$ permet d'éviter les défaut appelés "post-aliassage" [MITC 88] provenant d'une restitution incorrecte (cf. figure III.5). Le post-filtrage passe-bas dans ce cas correspond à un filtrage correct de restitution.

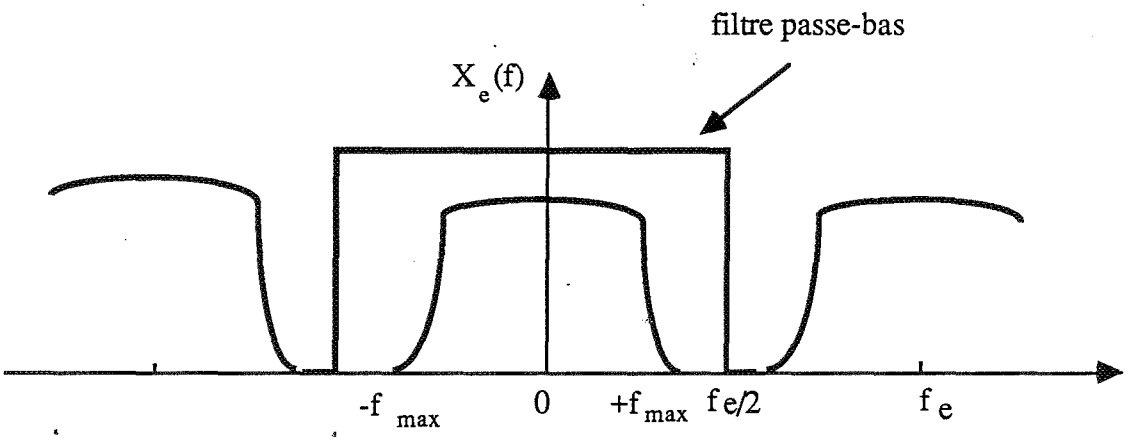


Figure III.5 : Filtrage de restitution pour éviter le "post-aliassage".

Conclusion :

Cette méthode est très simple, mais en général peu efficace ; elle nous permet de calculer la scène directement à la définition d'affichage puis d'adoucir les frontières entre les

objets. On perd beaucoup de hautes fréquences, tout à fait inutilement par rapport à une méthode de filtrage passe-bas avant échantillonnage. Par conséquent, l'image obtenue est floue, particulièrement dans le cas où les objets de l'image sont texturés. De plus, cette méthode ne permet pas de résoudre le problème des petits objets. Cette méthode est peu utilisée sauf dans le cas de scènes très simples, sans détail ni texture. L'image III.3 montre le résultat obtenu avec cette méthode (la fonction de Hamming est utilisée pour le filtrage). Le post-filtrage passe-bas appliqué sur l'image donne un résultat similaire à une défocalisation du moniteur (cf. chapitre II).

III.3.3. Méthodes de préfiltrage

On distingue deux classes de méthodes de préfiltrage : le préfiltrage discret d'une image virtuelle haute définition (sur-échantillonnée) et le préfiltrage analytique basé sur les propriétés géométriques des objets de la scène.

III.3.3.1 Sur-échantillonnage et préfiltrage discret

Cette méthode qui est simple à implémenter, générale et efficace dans la plupart des cas, peut être appliquée globalement ou localement. Avant de présenter cette méthode, nous allons préciser la signification des deux termes sur-échantillonnage et préfiltrage discret.

Sur-échantillonnage :

De façon rigoureuse et par référence au théorème d'échantillonnage (cf. annexe A), le terme sur-échantillonnage représente l'échantillonnage d'un signal analogique avec une fréquence d'échantillonnage strictement supérieure à la fréquence de Nyquist (égale à deux fois la fréquence maximale du signal).

Dans la suite, nous utiliserons de façon abusive le terme de "sur-échantillonnage" pour définir l'échantillonnage d'une scène sur une grille d'une définition beaucoup plus haute que la définition d'affichage.

Préfiltrage discret :

Le filtrage passe-bas appliqué à la scène sur-échantillonnée est un filtrage discret antirepliement (cf. chapitre II) qui limite les hautes fréquences et par conséquent évite l'aliassage.

III.3.3.1.1 Méthode de sur-échantillonnage global

La première étape consiste à sur-échantillonner l'image globalement, c'est à dire à générer l'image sur une grille de calcul haute définition. De façon théorique, la définition de la grille de calcul est donnée en fonction des hautes fréquences de la scène. Ce choix devra permettre de définir un nombre d'échantillons "suffisant" pour restituer "fidèlement" la scène. En pratique, on fixe la définition de façon expérimentale en fonction des détails de la scène.

Avec cette définition de la grille de calcul, chaque pixel de la grille d'affichage est divisé en plusieurs sous-pixels de calcul. Le nombre de divisions d'un pixel suivant l'axe des x (resp. des y) donne les deux rapports de sur-échantillonnage de la scène. Soit x_sur (resp. y_sur) le rapport de sur-échantillonnage selon x (resp. y) ; la définition de la scène numérique est alors donnée par :

$$D_s = (x\_sur).(y\_sur).D_a \qquad \text{(III.1)}$$

où D_a est la définition de la grille d'affichage.

Le choix de x_sur et y_sur influe directement sur les temps de calcul. Sur un plan théorique, les rapports de sur-échantillonnage doivent être très grands. En pratique, ils sont définis de manière à obtenir un compromis entre la qualité de l'image et le coût du calcul.

En général, le spectre fréquentiel de la scène haute définition contient pratiquement autant de hautes fréquences spatiales suivant x que suivant y, on pourra donc choisir x_sur et y_sur égaux. Dans ce cas, chaque pixel d'affichage est divisé en m^2 sous-pixels de calcul, *m* étant le rapport unique de sur-échantillonnage. La définition de la scène sur-échantillonnée est donnée par :

$$D_s = m^2.D_a \qquad \text{(III.2)}$$

La deuxième étape consiste à filtrer l'image sur-échantillonnée précédemment puis à la rééchantillonner à la définition d'affichage. Le filtre utilisé est un filtre passe-bas destiné à limiter le spectre fréquentiel de l'image avant l'affichage. Le filtrage dans le domaine fréquentiel étant exclu pour son temps de calcul élevé, on utilise un filtrage dans le domaine spatial en convoluant numériquement la fonction (RIF) du filtre passe-bas antirepliement avec l'image sur-échantillonnée (cf. algorithme II.1).

Le support du filtre passe-bas antirepliement utilisé est limité. Le choix du support, S_h, et de la fonction du filtre, h(x,y), résulteront d'un compromis entre le coût du calcul, la sauvegarde des hautes fréquences et la présence des effets d'aliassage dans l'image. Ainsi, la fonction et le support du filtre pourront varier suivant l'application et le périphérique d'affichage utilisé. On remarquera que la fonction et le support choisis peuvent être différents de ce que donnerait l'étude strictement théorique. On trouvera une présentation des fonction les plus utilisées au chapitre précédent.

Il n'existe pas de règles absolues pour choisir le support du filtre. Ainsi, Crow [CROW 77] propose un support tel que $S_h=\frac{2.D_s}{D_a}$ alors que, pour des raisons de simplicité, la solution la plus courante consiste à utiliser un support tel que $S_h=\frac{D_s}{D_a}$ avec en général une fonction normalisée de Fourier, c'est à dire une moyenne simple. L'emploi de cette fonction avec un support tel que $\frac{D_s}{D_a} = 2^i$ est particulièrement intéressant pour l'implémentation car l'opération de filtrage ne nécessite que 2^i additions et i décalages à droite par pixel d'affichage.

Le spectre fréquentiel de l'image sur-échantillonnée et préfiltrée est suffisamment limité pour que l'image puisse être rééchantillonnée à la définition de la grille d'affichage. La reconstitution (affichage) de l'image à partir de ces échantillons ne présente alors plus de phénomène d'aliassage perceptible si les rapports de sur-échantillonnage sont assez élevés.

Cette méthode est particulièrement bien adaptée à un affichage de l'image ligne par ligne. Nous avons utilisé le sur-échantillonnage global avec un filtrage local (filtrage dans les zones à problèmes) pour l'affichage de scènes polygonales 2D définies en haute définition avec un affichage par ligne de balayage [GHAZ 85].

Signalons ici un problème rencontré lors de l'affichage de scènes composées de segments et de polygones. L'affichage de segments, d'épaisseur théorique nulle, suppose toujours une définition explicite ou implicite de l'épaisseur. Il est généralement admis que l'épaisseur d'un segment est égale au côté d'un pixel. Cependant, cette définition est ambiguë dans le cas de sur-échantillonnage. En effet, les résultats seront différents suivant que l'on considère l'affichage sur la grille d'affichage ou sur la grille de calcul sur-échantillonnée. Si on prend pour épaisseur d'un segment un sous-pixel de la grille sur-échantillonnée, on obtiendra un segment très peu visible lors de l'affichage. Une solution consiste à définir les segments comme des polygones minces [CATM 78]. Pour notre part, nous avons choisi [GHAZ 85], afin de ne pas trop augmenter le volume des données, de garder la définition sous forme de segments et de fixer une épaisseur de segment constante,

généralement égale au côté d'un pixel sur la grille d'affichage. Cette épaisseur est prise en compte au fur et à mesure du traitement.

Les images III.4 et III.5 représentent les résultats obtenus à l'aide de cette méthode avec un support de filtre égal à $S_h = \frac{D_s}{D_a}$. Pour l'image III.4.a, x_sur= 1 et y_sur= 5, la scène est sur-échantillonnée et préfiltrée par une moyenne simple uniquement suivant l'axe des y. Comme l'on pouvait le prévoir théoriquement (cf. chapitre I), les marches d'escalier irrégulières sont estompées pour les pentes plutôt horizontales et apparentes pour les pentes plutôt verticales. Dans les images III.4.b et III.5 le préfiltrage est effectué à l'aide d'une fenêtre normalisée de Fourier (moyenne simple) avec les rapports de sur-échantillonnage x_sur = y_sur = 5. L'image III.6 qui est le zoom x5 de l'image III.5 montre l'effet du filtrage passe-bas qui est une transition plutôt continue entre les couleurs des segments et du fond. La comparaison des résultats obtenus sur l'image III.4.b (resp. III.5) et l'image III.2 (resp. III.1) confirme que l'augmentation de la définition apparente est liée à l'augmentation du nombre de niveaux d'intensité. Leler [LELE 80] montre qu'une primitive (stimulus) ainsi antialiassée apparaît au système œil-cerveau humain comme représentée avec une définition "infinie".

Conclusion :

La méthode de sur-échantillonnage global et préfiltrage est simple et générale ; elle permet de traiter les problèmes des petits objets et plus particulièrement l'effet de marches d'escalier irrégulières sans perdre inutilement la finesse (présence de hautes fréquences spatiales) de l'image. La généralité de la méthode est très intéressante pour éviter les cas particuliers comme des extrémités de segments et des sommets de polygones qui subsistent dans beaucoup d'autres méthodes. Par ailleurs, cette technique permet de traiter d'une façon simple des scènes variées, quelles que soient les entités qui les composent (segments, polygones, carreaux bilinéaires ou bicubiques,...).

Les inconvénients majeurs de cette méthode sont en général l'encombrement important de la mémoire pour le calcul de l'image virtuelle haute définition et le coût assez élevé de calcul dû essentiellement au sur-échantillonnage de toute la scène. Signalons que l'exploitation du balayage permet de substantielles économies de place mémoire.

III.3.3.1.2 Méthode de sur-échantillonnage local

Comme on vient de le voir, les inconvénients des méthodes de sur-échantillonnage global sont dus au calcul d'une image haute définition avant l'affichage de l'image finale.

Dans certains cas, on peut profiter des avantages du sur-échantillonnage sans subir ses inconvénients. En localisant les zones à sur-échantillonner on peut diminuer sensiblement le temps de calcul. La brosse antialiassée présentée dans [WHIT 83] est un exemple d'une méthode de sur-échantillonnage local. Dans cette méthode, seule la brosse, qui a une taille beaucoup moins importante que l'image entière, est calculée en haute définition. On peut trouver un exemple détaillé d'une méthode de "sur échantillonnage" local dans [MICH 87] et [PERO 88]. Certaines méthodes présentées dans les chapitres III et IV de ce mémoire (cf. paragraphes III.4.4, IV.4.4 et IV.5) sont quelques exemples intéressants du sur-échantillonnage local.

III.3.3.2 Méthodes géométriques et incrémentales

Les temps de calculs et encombrement mémoire du sur-échantillonnage nous conduisent à employer des méthodes géométriques et incrémentales. Le sur-échantillonnage est bien adapté à un algorithme d'affichage de la scène ligne à ligne. Nous allons maintenant aborder des méthodes de traitement des bords de polygones et des segments qui sont bien adaptées à l'affichage de la scène primitive par primitive (polygone ou segment). Les problèmes liés à l'affichage de la scène polygone par polygone seront décrits dans le paragraphe III.3.4. Les méthodes de traitement des segments et des bords de polygones décrites dans ce paragraphe sont des méthodes incrémentales basées sur les propriétés géométriques de ces éléments.

La méthode la plus courante pour afficher un segment de droite sur une mémoire d'image est l'algorithme incrémental de Bresenham [BRES 65]. Fondamentalement, cette méthode est basée sur l'interprétation de la variable de décision (erreur) d_B, qui est égale à d_1-d_2 (cf. figure III.6) ; d_B est une mesure de la distance entre le centre du pixel et le segment théorique. Ce paramètre d_B est proportionnel à la la valeur $2m(i+1)-2j-1+2c$ où (i,j) désigne les coordonnées du pixel courant et $y= mx+c$ est l'équation du segment à afficher. Si $d_B<0$, pour un segment situé dans le premier octant (une pente comprise entre 0 et 1), on affichera le pixel (i+1,j) comme appartenant au segment ; sinon, on prendra le pixel (i+1,j+1).

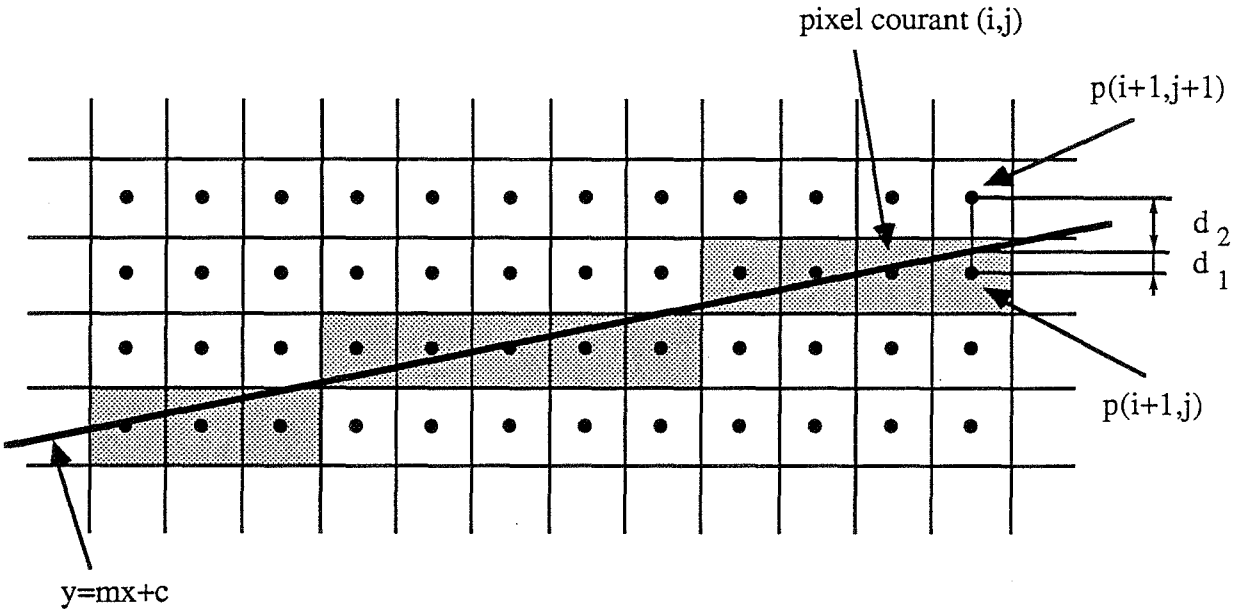


Figure III.6 : Affichage de segment par la méthode de Bresenham.

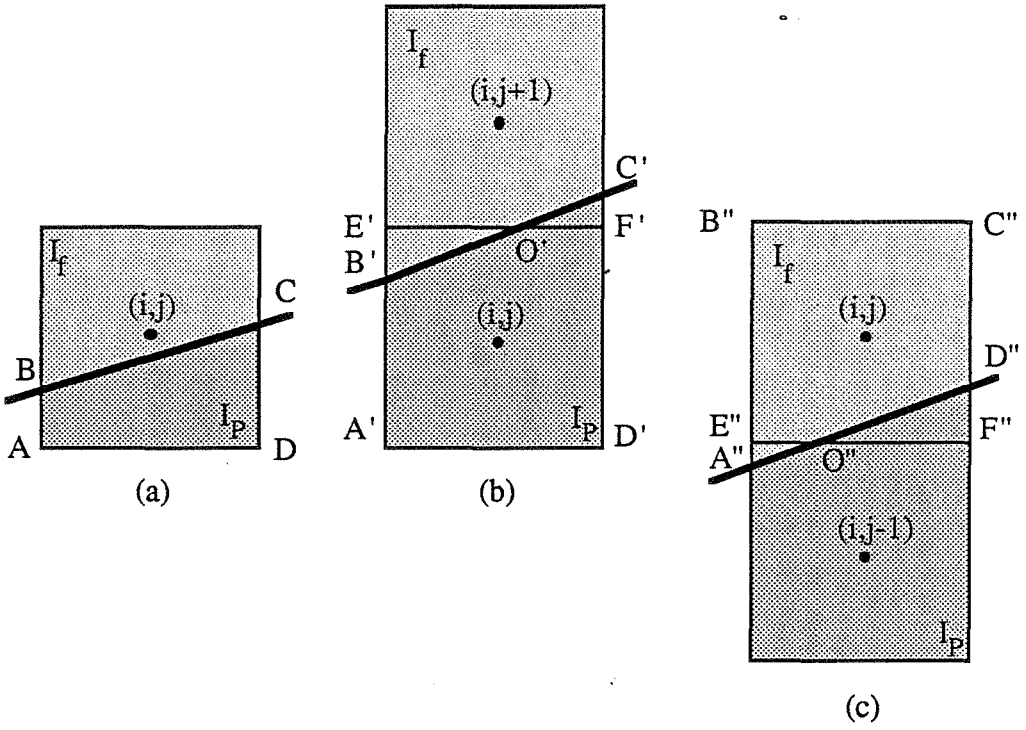
La variable d_B interprétée comme une notion de distance peut donner des informations géométriques pour déterminer la façon dont le segment (ou le bord de polygone) contribue à l'intensité des pixels voisins. Certaines modifications de l'évaluation de d_B permettent la génération de segments et de bords de polygones antialiassés d'une manière simple et incrémentale avec un temps de calcul très raisonnable.

III.3.3.2.1 Présentation de la méthode de Pitteway et Watkinson

La première méthode incrémentale basée sur l'algorithme de tracé de segment de Bresenham a été présentée en 1980 dans [PITT 80]. Cette méthode permet d'antialiasser les bords de polygones de façon très simple et, par conséquent, ses principes sont souvent repris pour des implémentations matérielles dans les systèmes graphiques. Pour étudier cette méthode, nous considérons les bords de polygones situés dans le premier octant, c'est à dire ceux dont l'orientation est comprise entre 0 et 45 degrés. L'extension aux éléments situés dans les autres octants est triviale.

La principale différence entre la méthode de Pitteway et Watkinson et celle de Bresenham réside dans l'interprétation du paramètre de décision. Bresenham utilise la valeur du paramètre de décision uniquement pour définir la position du pixel suivant. Pitteway et Watkinson définissent un nouveau paramètre de décision, d, qui permet à la fois de choisir la position d'un pixel et de calculer son intensité.

Nous allons maintenant présenter la méthode de calcul de l'intensité d'un pixel. Soient $I_p(x,y)$ et $I_f(x,y)$ les intensités des zones Z_p et Z_f délimitées par la droite d'équation y=mx+c. La méthode de [PITT 80] calcule l'intensité d'un pixel en fonction de la contribution de chacune des deux zones. L'évaluation de cette contribution dépend du type d'intersection entre le segment et le pixel. Pitteway et Watkinson distinguent deux cas, présentés sur les figures III.7.a et III.7.b, suivant le nombre d'intersections du segment avec les pixels d'une colonne donnée.



(a) Le segment intersecte un seul pixel de la colonne considérée.

(b) Le segment intersecte deux pixels (i,j) et (i,j+1) de la colonne considérée.

(c) Le segment intersecte deux pixels (i,j) et (i,j-1) de la colonne considérée.

Figure III.7 : Les trois cas d'intersection d'un segment avec la colonne i.

Dans le premier cas le segment n'intersecte qu'un seul pixel (i,j) de la colonne considérée (cf. figure III.7.a). L'aire de la portion du pixel (i,j) couverte par Z_p est égale à :

$$S_{ABCD} = \frac{AB+DC}{2} = \frac{m.(i - \frac{1}{2}) + c - (j - \frac{1}{2}) + m.(i + \frac{1}{2}) + c - (j - \frac{1}{2})}{2}$$

$$= m.i - j + \frac{1}{2} + c \qquad \text{(III.3)}$$

Dans le deuxième cas le segment intersecte deux pixels consécutifs (i,j) et (i,j+1) (cf. figure III.7.b). L'aire totale des pixels (i,j) et (i,j+1) couverte par Z_p est égale à :

$$S_{A'B'C'D'}=\frac{A'B'+D'C'}{2}= m.i - j+\frac{1}{2}+c \qquad \text{(III.4)}$$

On constate que les valeurs données par (III.3) et (III.4) sont égales à la valeur du paramètre de décision de l'algorithme de Bresenham augmentée de la valeur 1-m. On définit la nouvelle variable de décision, d, par :

$$d= m.i - j+\frac{1}{2}+c \qquad \text{(III.5)}$$

Avec cette définition, nous avons $0 \leq d \leq 1$.

On peut généraliser ce deuxième cas à un troisième représenté par la figure III.7.c. Dans ce cas le segment intersecte les deux pixels (i,j-1) et (i,j). Cette fois ci, l'aire totale des pixels (i,j-1) et (i,j) couverte par Z_f est égale à :

$$S_{A"B"C"D"}= \frac{(A"B"+D"C")}{2} = \frac{(j+\frac{1}{2})-m.(i-\frac{1}{2})-c+j+\frac{1}{2}-m.(i+\frac{1}{2})-c}{2}$$
$$= j-m.i-c+\frac{1}{2} = 1-d \qquad \text{(III.6)}$$

L'intensité finale attribuée au pixel (i,j) sera déterminée dans les trois cas en fonction des aires couvertes par Z_f et Z_p, calculées précédemment.

$$I_{final}(i,j)= d.I_p(i,j)+(1-d).I_f(i,j) = d.[I_p(i,j)-I_f(i,j)]+I_f(i,j) \qquad \text{(III.7)}$$

Il faut noter que dans les deux derniers cas (figures III.7.b et III.7.c), seuls les pixels (i,j) seront traités par (III.7).

Cette méthode est représentée par l'algorithme suivant pour des bords de polygones situés dans le premier octant. La généralisation aux autres octants est triviale.

Algorithme III.1, la méthode de Pitteway et Watkinson :

```
{
 entiers : If,Ip,Ifinal,i,j,ifin;
 réels : xdébut,ydébut,xfin,yfin,m,w,c,d;

 /* Initialisations */
 m:= (yfin-ydébut)/(xfin-xdébut);
 w:= 1-m; d:= c+0.5;
 i:= arrondi de (xdébut); ifin:= arrondi de (xfin);
```

```
j:= arrondi de [y_début+m.(i-x_début)];

tant que i≤i_fin faire
{ /* calcul d'intensité et affichage du pixel de bord */
  lire I_f(i,j);
  I_final(i,j):= arrondi de (d.[I_p(i,j)-I_f(i,j)])+I_f(i,j);
  afficher I_final(i,j);
  i:= i+1;
  si d<w alors d:= d+m
  sinon
   { j:= j+1;
     d:= d-w;
   }
}
}
```

Conclusion :

La méthode présentée dans ce paragraphe est une méthode incrémentale simple et efficace pour antialiasser les bords de polygones (les segments peuvent être traités avec cette méthode s'ils sont considérés comme des polygones minces).

Comme Pitteway et Watkinson l'ont remarqué, il faut traiter les sommets des polygones comme des cas particuliers, si l'on veut obtenir un traitement correct. De plus, un pixel peut être intersecté par plusieurs zones (figure III.8) et le traitement donné par l'algorithme précédent ne correspond pas à un calcul d'intensité correct pour ce pixel. Le problème des petits objets ne peut pas être traité par cette méthode. Il existe les même problèmes pour toute autre méthode de ce type.

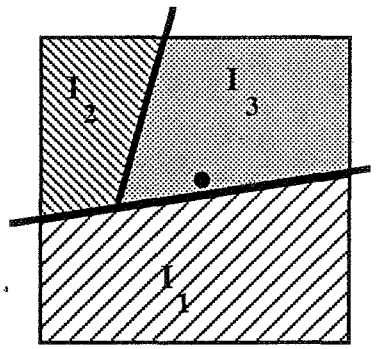


Figure III.8 : Problème rencontré pour un pixel intersecté par plusieurs polygones.

La formule du calcul d'intensité donnée par (III.7) correspond à une opération de préfiltrage par une moyenne simple (on prend en compte les pourcentages des surfaces

occupées par le polygone et par le fond) au lieu d'une moyenne pondérée qui correspond à un filtrage plus sophistiqué. Si $I_f(i,j)$ n'est pas connu a priori, il faut ajouter le temps de lecture sur la mémoire d'image qui peut être important dans certains cas par rapport au temps total de calcul.

Dans les cas des figures III.7.b et III.7.c, le fait de ne traiter que le pixels (i,j) avec la formule donnée par (III.7) ne correspond pas à un traitement très précis, ni pour les pixels (i,j), ni pour les pixels (i,j+1) et (i,j-1). De plus, le traitement d'un seul pixel par colonne peut aggraver l'effet "d'ondulation" ("twisted rope") constaté à des degrés variables dans toutes les autres méthodes. Cet effet est dû à la faible définition d'affichage (cf. paragraphe III.3.5). Nous avons proposé quelques modifications de cette méthode pour un calcul d'intensité plus précis de ces deux cas dans [GHAZ 85]. L'image III.7 représente le résultat du traitement appliqué avec les modifications apportées à la méthode de Pitteway et Watkinson proposées dans [GHAZ 85].

III.3.3.2.2 Autres méthodes géométriques

Une deuxième méthode géométriques et incrémentale basée sur les principes de l'algorithme de Bresenham a été publiée par Gupta et Sproull [GUPT 81]. Cette méthode permet l'antialiassage de segments et de bords de polygones. Comme la méthode précédente, Gupta et Sproull utilisent un nouveau paramètre de décision qui permettra de choisir les pixels à afficher et de leur affecter une valeur d'intensité. Alors que dans la méthode de Bresenham, le paramètre de décision fournit une évaluation de la distance entre le centre de pixel et l'axe du segment suivant la direction verticale, la méthode de [GUPT 81] calcule la distance entre le centre du pixel et l'axe du segment suivant une direction perpendiculaire à ce dernier.

Pour des raisons de simplicité, la fonction (RIF) du filtre passe-bas choisie est une fonction conique, la symétrie circulaire du filtre facilitant le calcul d'intensité. Par ailleurs, cette fonction est normalisée. Le centre du filtre est situé au centre d'un pixel. Le choix du rayon, *r*, de la base du filtre influe sur l'adoucissement des bords. Un *r* grand correspond à un adoucissement important alors qu'un *r* petit implique le maintien des hautes fréquences. On notera que cette remarque n'est pas spécifique de ce type de filtre.

Pour effectuer la convolution, cette méthode utilise une table de conversion ayant comme indice la distance entre le centre du pixel et le bord de polygone. Dans le cas des segments, la table de conversion a deux indices : la distance entre le centre de pixel et l'axe de segment et l'épaisseur du segment.

Une autre méthode géométrique intéressante basée sur les mêmes principes de filtrage est celle de Turkowski [TURK 82]. Comme la méthode de Gupta et Sproull [GUPT 81], le calcul de la convolution entre la fonction du filtre passe-bas et la scène est lié à un calcul de distance. Cette distance est utilisée comme indice dans une table de conversion donnant la valeur de la convolution.

Parmi d'autres méthodes géométriques et incrémentales d'antialiassage de traits et de bords de polygones, on peut citer la méthode de Fujimoto et Iwata [FUJI 83] utilisant un filtre passe-bas ayant comme fonction normalisée une fenêtre de Fourier dont le support est ajustable. Comme les autres méthodes géométrique et incrémentales, le calcul des coordonnées des pixels à afficher et l'évaluation de l'intensité de ces pixels sont faits simultanément.

On peut trouver une étude détaillée et comparative des différentes méthodes géométriques et incrémentales dans [GHAZ 85].

III.3.4 Problèmes liés aux techniques d'affichage

Après avoir présenté les différentes méthodes de traitement de l'aliassage, nous allons décrire plusieurs problèmes qui peuvent être rencontrés et qui sont relatifs à la technique d'affichage employée. On peut distinguer principalement deux techniques d'affichage :

- affichage de la scène primitive par primitive, ce qui correspond à un affichage polygone par polygone pour une scène polygonale ;

- affichage ligne par ligne.

Affichage polygone par polygone :

Cette technique est bien adaptée aux méthodes géométriques et incrémentales pour l'antialiassage. Elle consiste à afficher les polygones constituant la scène les uns après les autres. Lors de l'affichage de l'un des polygones, on distinguera les pixels entièrement situés à l'intérieur du polygone qui seront affichés sans précaution et les pixels situés sur la frontière qui devront faire l'objet d'un traitement.

Le traitement d'un pixel (x,y) de bord est réalisé de la façon suivante :
Soient :

- p(x,y) l'intensité du point (x,y) pour le polygone p ;

- f(x,y) l'intensité du fond au point (x,y), qui est généralement obtenue par lecture du pixel ;
- c(x,y) la contribution de p à l'intensité du pixel (x,y) ; cette valeur est définie par la méthode d'antialiassage utilisée.

L'intensité calculée du point (x,y) est alors obtenue par une interpolation linéaire :

$$I(x,y)= c(x,y).p(x,y)+[1-c(x,y)].f(x,y) \qquad \text{(III.8)}$$

L'affichage de deux polygones adjacents (cf. figure III.9) pose alors un problème particulier. Soient p_1 et p_2 ces deux polygones et soit AB le bord communs à p_1 et p_2 (leur frontière). Supposons que p_1 soit affiché en premier, sans tenir compte de p_2 qui sera affiché ensuite. Le traitement d'antialiassage est donc appliqué une première fois aux pixels appartenant à AB :

$$I_1(x,y)= c(x,y).p_1(x,y)+[1-c(x,y)].f(x,y) \qquad \text{(III.9)}$$

Ensuite, la frontière AB est traitée une deuxième fois lors de l'affichage du polygone p_2. L'intensité du fond lue au point (x,y) est alors la valeur $I_1(x,y)$ issue de la première phase de traitement.

$$\begin{aligned} I'_{final}(x,y)&= c(x,y).I_1(x,y) + [1-c(x,y)].p_2(x,y) \\ &= c^2(x,y).p_1(x,y)+[1-c(x,y)].p_2(x,y)+c(x,y).[1-c(x,y)].f(x,y) \end{aligned} \qquad \text{(III.10)}$$

Cependant, la valeur d'intensité correcte d'un pixel (x,y) appartenant à la frontière AB devrait être :

$$I_{final}= c(x,y).p_1(x,y)+[1-c(x,y)].p_2(x,y) \qquad \text{(III.11)}$$

L'erreur absolue induite est égale à :

$$\begin{aligned} I_{erreur}&= |I_{final}(x,y)-I'_{final}(x,y)| \\ &= c(x,y).[1-c(x,y)].|p_1(x,y)-f(x,y)| \end{aligned} \qquad \text{(III.12)}$$

donc

$$I_{erreur\text{-}max} = 0.25|p_1(x,y) - I_f(x,y)| \qquad \text{(III.13)}$$

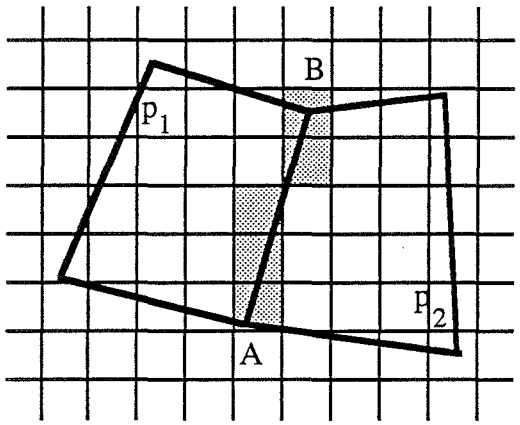


Figure III.9 : Affichage successif de deux polygones adjacents.

De façon générale, un calcul d'intensité correct nécessite de connaître toute l'information au moment du traitement. Dans le cas précédent, ce problème pourrait être résolu en traitant la frontière AB uniquement après l'affichage de p_2. Il faut signaler qu'une solution, plus pratique, consiste à lire l'intensité du pixel voisin du pixel (x,y) sur p_1 qui a souvent une valeur égale, ou très proche, de $p_1(x,y)$. La même technique peut être utilisée pour traiter les bords d'un polygone après son affichage. Dans ce cas, on lit l'intensité du pixel du bord qui correspond à l'intensité exacte du polygone en ce point et la lecture du pixel voisin sur le fond nous donnera l'intensité correcte, ou une valeur très proche, du fond en ce point. Cependant cette solution ne résout pas le problème des sommets de polygones qui devront être traités comme des cas particuliers.

Affichage ligne par ligne :

Avec cette méthode, l'image est affichée ligne par ligne. Le traitement d'antialiassage d'un pixel (x,y) situé sur la frontière entre deux polygones p_1 et p_2 (cf. figure III.10) est donc réalisé en connaissant les valeurs d'intensité de p_1 et p_2 au point (x,y) ainsi que les valeurs d'intensité de tous les points voisins situés à l'intérieur du support du filtre passe-bas utilisé. Ainsi, contrairement à la méthode précédente, toute l'information nécessaire est connue lors du traitement de chaque pixel. Un traitement correct est alors possible. Cette technique d'affichage est bien adaptée au sur-échantillonnage.

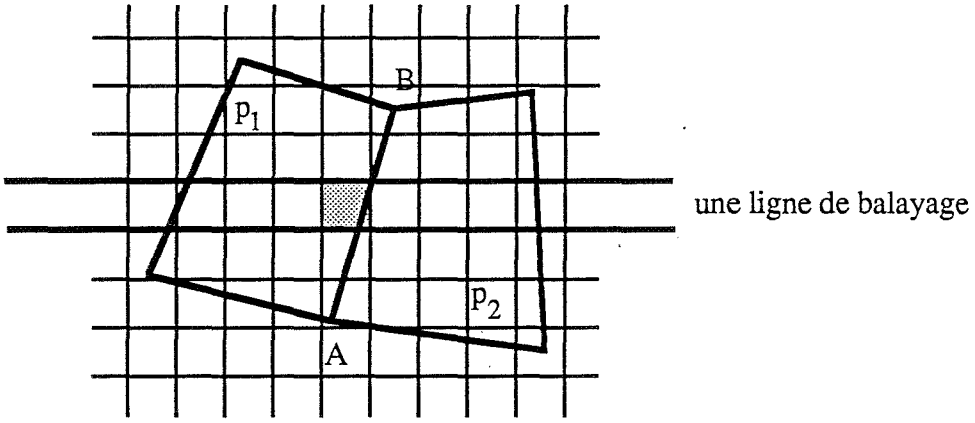


Figure III.10 : Affichage par ligne de balayage.

III.3.5 Conclusion

Pour conclure en ce qui concerne l'antialiassage 2D, rappelons que le post-filtrage n'est pas une méthode acceptable (sauf pour de rares cas). Par contre, le préfiltrage est une solution correcte. Parmi les méthodes de préfiltrage, le sur-échantillonnage est la méthode la plus générale. Elle permet de traiter des entités quelconques ; de plus les problèmes d'extrémités de segments, de sommets de polygones et de petits objets sont pratiquement résolus. Par contre, l'affichage des segments suppose la définition d'une épaisseur non nulle ; un segment doit généralement être traité soit comme un polygone soit comme un cas particulier. Les méthodes basées sur un sur-échantillonnage de la scène demandent en général des temps de calcul importants.

Les méthodes géométriques et incrémentales sont d'autres méthodes de préfiltrage. Pour des raisons de simplicité, ces méthodes sont parmi les plus utilisées, en particulier pour les implémentations matérielles. Pour ces méthodes, les temps de calcul nécessaires sont assez faibles. La résolution des cas particuliers (extrémités de segments et sommets de polygones) nécessite un traitement à part. Ceci peut poser des problèmes pratiques, par exemple des tables de conversion de très grande taille dans la méthode de Gupta et Sproull (cf. [GUPT 81]) ou des calculs assez importants dans une méthode géométrique telle que celle de [FEIB 80] qui n'est pas une méthode incrémentale.

Les images traitées par des méthodes de préfiltrage sont généralement d'une qualité acceptable. On peut cependant signaler un problème. Il s'agit du phénomène "d'ondulation" présent sur les bords des différentes entités d'une image. Ce problème, qui peut subsister quelle que soit la méthode de préfiltrage appliquée, est essentiellement dû à la définition de la mémoire de trame. Avec une définition de l'ordre de 1024x1024 pixels, ce problème est

pratiquement supprimé (d'après [FUJI 83]). Cependant, il peut être très atténué sur des mémoires de trame de plus faible définition en utilisant un support de filtre plus grand au prix d'une image plus floue.

L'intensité par unité de distance est variable selon la pente du segment [FOLE 82]. Considérons les 2 segments s_1 de pente 0 et s_2 de pente 1 (cf. figure III.11). Si on note L(s) la longueur du segment s et I son intensité, on a $L(s_2)= \sqrt{2}.L(s_1)$. La densité de l'intensité pour s_1 est I et celle de s_2 est $\frac{I}{\sqrt{2}}$ car les deux segments contiennent le même nombre de pixels affichés. Ce problème est visible sur des segments de pentes différentes dans une image. En principe, les algorithmes d'antialiassage traitent ce problème en donnant une densité constante au segment. Cependant une méthode comme celle de Fujimoto et Iwata [FUJI 83] qui garde l'intensité totale d'une colonne constante (cf. [FUJI 83]) ne va pas traiter ce problème et la différence de luminosité sur les segments de pentes différentes est visible.

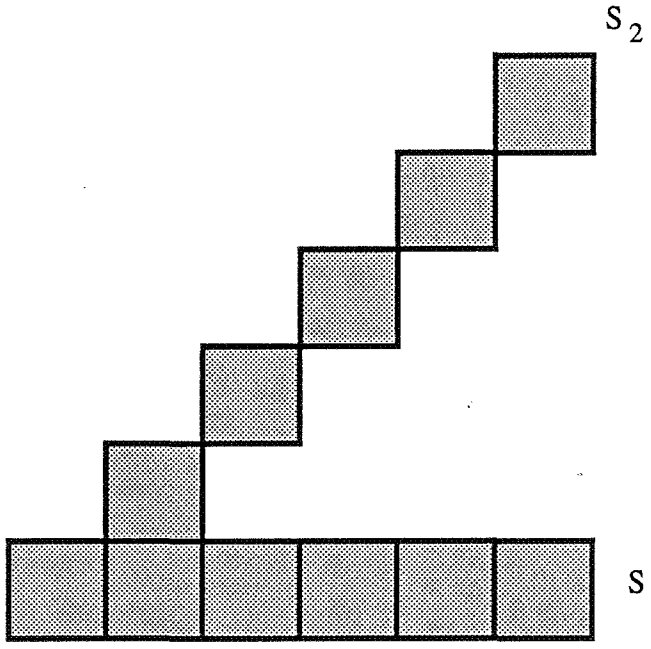


Figure III.11 : La différence de la densité de l'intensité lumineuse entre un segment horizontal s_1 (de pente 0) et un segment s_2 diagonal (de pente 1).

Considérons l'affichage de segments sur une grille de pixels. Si les extrémités des segments sont recalées sur la grille d'affichage, alors l'animation des images peut créer des sauts de segments assez visibles dus aux discontinuités dans les mouvements. La possibilité d'utiliser des extrémités de segments non recalées sur la grille d'affichage évite les phénomènes de saut de segments et diminue les problèmes relatifs aux extrémités des

segments et aux sommets de polygones. Il est donc conseillé de traiter des segments dont les extrémités n'ont pas été recalées. Le même problème se rencontre avec l'affichage de polygones, mais la taille des polygones rend le décalage moins sensible.

Les images obtenues par les différentes méthodes de préfiltrage (cf. [GHAZ 85]) montrent que pratiquement la même qualité peut être obtenue à l'issue d'un traitement par une méthode simple (par exemple [PITT 80]) que par une méthode coûteuse comme le sur-échantillonnage. Par conséquent, si le but est uniquement le traitement du phénomène très courant de marches d'escalier, il est préférable d'utiliser une méthode simple et peu coûteuse. Par contre, s'il est nécessaire de traiter les petits objets ou si l'on souhaite un traitement très précis sur les sommets de polygones et les extrémités de segments, une méthode plus sophistiquée doit être employée.

III.4 Méthodes d'antialiassage 3D

Dans le paragraphe III.3, nous avons étudié le problème de l'antialiassage des objets géométriques. Diverses méthodes d'antialiassage ont été exposées dans un contexte 2D sans se préoccuper de l'élimination des parties cachées et des problèmes y afférents. Or l'élimination des parties cachées est un élément important des techniques de visualisation de scènes 3D sans lequel l'affichage correct des images n'est pas possible. Certains algorithmes d'élimination des parties cachées sont difficilement compatibles avec la nécessité de disposer des informations indispensables pour un traitement d'antialiassage. Dans ce cas, un traitement d'antialiassage exact est difficilement réalisable. En particulier, cette incompatibilité est importante pour les techniques de visualisation les plus simples et par conséquent les plus répandues. Dans le cas d'autres méthodes de visualisation, le traitement d'antialiassage classique devient très coûteux en temps de calcul. Cependant, si on accepte quelques approximations ou compromis, on peut envisager des solutions pratiques pour un traitement d'antialiassage d'une qualité acceptable.

Dans ce qui suit, nous allons étudier brièvement les problèmes d'application d'un traitement d'antialiassage, de façon générale, à quatre des techniques de visualisation les plus courantes des scène 3D : le tri en profondeur ("depth sort"), le balayage ligne par ligne ("scan line"), la subdivision de surface ("area subdivision") et le lancer de rayons ("ray tracing"). Etant données l'importance de la méthode du tampon de profondeur ("z-buffer") parmi les techniques de visualisation et les difficultés d'application d'un antialiassage dans ce cas, les problèmes liés à cette dernière méthode feront l'objet d'une étude détaillée dans le chapitre suivant.

III.4.1 Antialiassage et tri en profondeur

La méthode du tri en profondeur, appelée souvent l'algorithme du peintre, publiée par Newell, Newell et Sancha [NEWE 72], génère l'image polygone par polygone. Très brièvement, on peut résumer la méthode dans le cas général de la manière suivante : il faut trier tous les polygones qui composent la scène dans l'ordre croissant selon leur profondeur maximale (ceci suppose que le problème des polygones pénétrants soit résolu auparavant). Après avoir obtenu ainsi une liste de priorité des polygones, on peut les afficher successivement, les polygones les plus éloignés de l'œil étant affichés les premiers. Ce qui est équivalent à l'affichage d'une scène 2.5D d'arrière en avant.

Les bords d'un polygone, P_j, sont traités lors de son affichage. En général les pixels (x,y) de P_j concernés par le fond peuvent être traités en connaissant l'intensité du fond au voisinage de ce point. En fait le fond est constitué par l'ensemble des polygones, y compris le fond d'origine, affichés avant l'affichage de P_j.

Des problèmes de traitement d'antialiassage se présentent lorsque deux polygones P_j et P_i sont adjacents. Dans ce cas, les pixels appartenant au bord en commun sont traités une première fois lors de l'affichage de P_i et une deuxième fois lors de l'affichage de P_j. Le calcul de l'intensité correcte pour les pixels concernés par ce bord en commun pose alors un problème. Ce problème est le même que celui rencontré lors de l'affichage des polygones adjacents étudié dans le cas 2D (cf. paragraphe III.3.4).

La solution théorique consiste à tenir à jour une liste de recouvrement pour chaque pixel de bord. Pratiquement, cette solution est difficilement envisageable si le nombre de pixels de bord est assez important. Comme nous l'avons remarqué dans le paragraphe III.3.4, une solution pratique consiste à lire l'intensité du pixel voisin du pixel de bord.

Cette méthode d'élimination des parties cachées convient à un antialiassage des scène polygonales par une méthode géométrique et incrémentale.

III.4.2 Antialiassage et la méthode de balayage ligne par ligne

Dans la méthode précédente d'élimination des parties cachées, l'image est affichée polygone par polygone. Donc, lors de l'affichage d'un polygone, seules les informations propres à ce polygone sont disponibles. On a constaté précédemment que ces informations ne sont malheureusement pas suffisantes pour un traitement correct d'antialiassage. Nous étudions ici un autre type de méthode d'élimination des parties cachées s'appuyant sur un

balayage de ligne. Parmi les différents algorithmes de cette classe, celui de Watkins traitant des scènes polygonales [WATK 70] est un des plus connus. On peut résumer les principes de cette méthode de la façon suivante. Considérons l'intersection de chaque polygone de la scène avec un plan xy contenant la ligne de balayage. On obtient un ensemble de segments d'intersection qui sont ensuite triés en x suivant leur première extrémité. Un intervalle de la ligne de balayage entre deux extrémités successives s'appelle un empan. Si dans un empan il n'y a aucun segment, le fond est visible ; s'il y a un seul segment c'est le polygone contenant ce segment qui est visible. Dans le cas d'un empan avec plusieurs segments, une comparaison des valeurs de Z est nécessaire pour déterminer le polygone visible.

Dans cette méthode, l'image est calculée et affichée ligne à ligne. Contrairement à la méthode précédente, lors de l'affichage d'une ligne nous disposons de toutes les informations de la scène affectant cette ligne. Ceci rend possible un traitement d'antialiassage correct. Une solution bien adaptée pour effectuer le traitement d'antialiassage est la mise en œuvre d'une méthode de sur-échantillonnage qui peut être global ou local.

III.4.3 Antialiassage et subdivision de surface

L'algorithme de Warnock [WARN 69] est un exemple d'algorithme de cette classe. L'algorithme est basé sur une subdivision récursive de l'image en quatre sous-images rectangulaires. Le test d'arrêt de la subdivision est, soit l'absence d'ambiguïté au niveau des polygones visibles dans un rectangle, soit la taille du rectangle devenue sensiblement égale à celle d'un pixel.

Comme dans la méthode précédente, on dispose de toutes les informations nécessaires au moment de la subdivision. De plus, l'algorithme est naturellement bien adapté à un traitement d'antialiassage par une méthode de sur-échantillonnage. On peut effectuer le sur-échantillonnage en continuant la subdivision jusqu'au niveau d'un sous-pixel dans les endroits où cela est nécessaire.

III.4.4 Antialiassage et lancer de rayons

L'algorithme de lancer de rayons introduit par Whited en 1980 [WHIT 80] est considéré à l'heure actuelle comme la technique la plus performante pour produire les images de synthèse les plus réaliste en simulant les effets optiques tels que les ombres portées et la transparence. L'algorithme de lancer de rayons est simple et il suit le trajet inverse des rayons de lumière pour produire les différents effets optiques. Les principes du lancer de rayons peuvent être résumés par l'algorithme suivant :

Algorithme III.2, la méthode de Whited :

```
{
 pour chaque pixel (x,y) de l'écran faire
   {
      lancer un rayon R du point de vue vers la scène en passant par (x,y);
      trouver l'intersection de R avec toutes les entités de la scène et choisir l'intersection la
      plus proche I;
      lancer des rayons de I vers les sources lumineuses;
      calculer la couleur du pixel (x,y) en fonction de l'éclairement (au sens large) de I;
   }
}
```

Signalons que le mot éclairement est utilisé au sens large dans l'algorithme et qu'il faut tenir compte des rayons réfléchis dans le cas des surfaces réfléchissantes. On constate que le lancer de rayons est un algorithme d'échantillonnage de points et comme tout autre processus de ce type, il est naturellement susceptible d'engendrer de sérieux problèmes d'aliassage. Etant donné que le lancer de rayons est employé pour générer les images les plus réalistes, l'antialiassage dans ce cas devient un problème primordial.

La première méthode d'antialiassage pour le lancer de rayons est proposée dans [WHIT 80]. Dans cette méthode, qui sert de base pour d'autres méthodes, l'antialiassage est inclus comme partie intégrale des calculs de visibilité. Pour trouver les pixels à problème (afin de procéder à antialiassage par un sur-échantillonnage local) [WHIT 80] utilise un modèle de grille de pixels qui est avantageusement utilisé dans d'autres approches comme celles décrites dans [CATM 74] ou [DUFF 85]. Dans ce modèle de grille, un pixel d'affichage est considéré comme un carré dont les coins sont quatre points d'échantillonnage. Par conséquent, le nombre des échantillons est augmenté d'une ligne et d'une colonne supplémentaires par rapport à une grille d'affichage classique pour laquelle le centre d'un pixel est un point d'échantillonnage (cf. figure II.1). Pour un pixel de la grille d'affichage, si les intensités calculées aux quatre coins sont presque équivalentes et si aucun petit objet n'est détecté dans la région correspondant au pixel en question, il est considéré comme uniforme et la moyenne des quatre intensités sera utilisée pour l'intensité du pixel. Dans le cas contraire, le pixel sera récursivement subdivisé en quatre et l'algorithme procède de la même manière jusqu'à ce que la région devienne uniforme ou que la limite de la subdivision soit atteinte. La détection d'un petit objet est réalisée en imposant une sphère englobante d'un rayon minimal défini en fonction de sa distance au point de vue afin d'assurer l'intersection de celle-ci avec au moins un rayon. Dans le cas d'intersection d'un rayon avec une telle sphère, les quatre pixels voisins de celui-ci seront subdivisés. L'auteur signale que cet algorithme est susceptible de poser des problèmes d'antialiassage pour des rayons réfléchis (secondaires) issus de surfaces gauches.

La méthode de [WHIT 80] utilisant un sur-échantillonnage local est une méthode classique intéressante qui peut être améliorée avec quelques légères modifications. En fait, l'idée de tester les quatre coins d'un pixel afin de détecter les pixels non-uniformes est intéressante. Cependant, le test des intensités des quatre coins d'un pixel, qui nécessite un seuil de tolérance fixé a priori, n'est pas toujours fiable, en particulier dans le cas des objets portant une texture. De plus, l'application d'un filtrage passe-bas autre que la moyenne simple utilisée donnera de meilleurs résultats. En séparant le problème d'aliassage des zones texturées (cf. chapitre V) des autres problèmes d'aliassage (marches d'escalier sur les contours et les petits objets), nous proposons quelques modifications qui sont résumées par l'algorithme suivant.

Algorithme III.3, une méthode d'antialiassage pour le lancer de rayons :

{
pour chaque pixel (x,y) de l'écran **faire**
{
lancer les rayons R_i ($1 \leq i \leq 4$) du point de vue vers la scène en passant par les coins du pixel qui n'ont pas déjà été pris en compte;
trouver les intersections de chaque R_i avec toutes les entités de la scène et choisir l'intersection la plus proche I_i;
lancer des rayons de chaque point I_i vers les sources lumineuses;
déterminer l'éclairement de chaque point I_i;
si (tous les points I_i appartiennent à la même entité **et** ils ont le même éclairement (au sens large) **et** aucun petit objet dans la région correspondante n'y est détecté) **alors** le pixel (x,y) est uniforme
sinon diviser le pixel (x,y) en sous pixels afin d'appliquer un filtrage passe-bas;
}
}

Parmi les autres méthodes d'antialiassage avec l'algorithme de lancer de rayons, on peut citer les suivantes :

La méthode de [AMAN 84] qui utilise le tracé de cônes ; par conséquent, elle évite l'échantillonnage de points qui pose des problèmes d'aliassage mais le problème de l'intersection des cônes et des entités composant la scène rend cette méthode complexe.

Les méthodes exposées dans [DIPP 85], [LEE 85], [COOK 86], [MITC 87] et [PAIN 89] sont des exemples d'utilisation de l'échantillonnage stochastique (cf. chapitres I et II) pour l'antialiassage avec le lancer de rayons. Pour ces méthodes, l'échantillonnage stochastique est en général combiné avec un sur-échantillonnage. En fait, l'utilisation directe des méthodes de sur-échantillonnage est souvent trop coûteuse avec le lancer de rayons car elles exigent une plus grande définition pour la grille de calcul. De plus, les méthodes

d'échantillonnage stochastique permettent plus facilement la simulation des effets optiques spéciaux comme les pénombres, la translucidité, la profondeur de champ, la brillance et le flou de bougé ("motion blur") afin d'atteindre un plus grand degré de réalisme (cf. [COOK 86]).

La méthode proposée dans [ARGE 88] est un antialiassage à l'aide d'un sur-échantillonnage local dans le cas du lancer de rayons utilisant une modélisation par arbre de construction ("CSG"). La détection des pixels à problème est faite en testant chaque pixel et ses huit voisins. Les pixels détectés seront subdivisés récursivement en neuf et l'uniformité de chaque sous-pixel sera détectée à nouveau de la même manière que pour un pixel d'affichage. La méthode telle qu'elle est exposée dépend de l'arbre de construction. Cette méthode sépare l'antialiassage des textures des autres cas et l'utilisation de l'une des méthodes employés dans le cas de textures (cf. chapitre V) est suggérée. La comparaison donnée dans [ARGE 88] sur un exemple montre une diminution du temps de calcul par rapport à l'algorithme de [WHIT 80].

Références du chapitre III

[AMAN 84] : J. AMANATIDES,
"Ray Tracing with Cones", Computer Graphics, vol. 18, No. 3, July 1984, pp. 129-145.

[ARGE 88] : J. ARGENCE,
"Antialiasing for Ray Tracing Using CSG Modeling", CG International'88, New Trends in Computer Graphics, Springer Verlag, 1988, pp. 199-208.

[BRES 65] : J. BRESENHAM,
"Algorithm for Computer Control of a Digital Plotter", IBM Syst. J. 4, 1965, pp. 25-30.

[CATM 74] : E. CATMULL,
"A Subdivision Algorithm for Computer Display of Curved Surfaces", ph.D. thesis, University of Utah, December 1974.

[CATM 78] : E. CATMULL,
"A Hidden-Surface Algorithm with Anti-Aliasing", Computer Graphics, vol. 12, No. 3, August 1978, pp. 6-11.

[COOK 86] : R. COOK,
"Stochastic Sampling in Computer Graphics", ACM Transactions on Graphics, vol. 5, No. 1, January 1986, pp. 51-72.

[CROW 77 : F. CROW,
"The Aliasing Problem in Computer Generated Shaded Images", Comm. ACM, vol. 20, No. 11, November 1977, pp. 799-805.

[DIPP 85] : M. DIPPE and E.WOLD,
"Antialiasing Through Stochastic Sampling", Computer Graphics, vol. 19, No. 3, July 1985, pp. 69-78.

[DUFF 85] : T. DUFF,
"Compositing 3-D Rendered Images", Computer Graphics, vol. 19, No. 3, July 1985, pp. 41-43.

[FEIB 80] : E. FEIBUSH, M. LEVOY and R. COOK,
"Synthetic Texturing Using Digital Filters", Computer Graphics, vol. 14, No. 3, July 1980, pp. 294-301.

[FOLE 82] : J. FOLEY and A. VAN DAM,
Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, 1982, page 438.

[FUJI 83] : A. FUJIMOTO and K. IWATA,
"Jag-Free Images on Raster Displays", IEEE Computer Graphics and Applications, vol. 3, No. 9, December 1983, pp. 26-34.

[GHAZ 85] : D. GHAZANFARPOUR,
"Synthèse d'images et antialiassage", Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure des Mines, Saint-Etienne, novembre 1985, pp. 72-114.

[GUPT 81] : S. GUPTA and R.F. SPROULL,
"Filtering Edges for Gray-Scale Displays", Computer Graphics, vol. 15, No. 3, August 1981, pp. 1-5.

[LANE 80] : J. LANE, L. CARPENTER, T. WHITED and J. BLINN,
"Scan-line Methods for Displaying Parametrically Defined Surfaces", Communications of the ACM, vol. 25, No. 1, 1980, pp. 23-34.

[LEE 85] : M. LEE,
"Statistically Optimized Sampling for Distributed Ray Tracing", Computer Graphics, vol. 19, No. 3, July 1985, pp. 61-67.

[LELE 80] : W. LELER,
"Human Vision, Anti-aliasing, and the Cheap 4000 Display", Computer Graphics, vol. 14, No. 3, July 1980, pp. 308-313.

[MICH 87] : D. MICHELUCCI,
"Les représentations par les frontières : quelques constructions ; difficultés rencontrées", Thèse de Doctorat, Université de Saint-Etienne et Ecole Nationale Supérieuer des Mines de Saint-Etienne, novembre 1987, pp. 44-47.

[MITC 87] : D. MITCHELL,
"Generating Antialiased Images at Low Sampling Densities", Computer Graphics, vol. 21, No. 4, July 1987, pp. 65-72.

[MITC 88] : D. MITCHELL and A. NETRAVALI,
"Reconstruction Filters in Computer Graphics", Computer Graphics, vol. 22, No. 4, August 1988, pp. 221-228.

[NEWE 72] : M. NEWELL, R. NEWE and T. SANCHA,
"A New Approch to Shaded Picture Problem", Proc. ACM Nat. Conf., 1972, p. 443.

[PAIN 89] : J. PAINTER and K.SLOAN,
"Antialiased Ray Tracing by Adaptative Progressive Refinement", Computer Graphics, vol. 23, No. 3, July 1989, pp. 281-288.

[PERO 88] : B. PEROCHE, J. ARGENCE, D. GHAZANFARPOUR et D. MICHELUCCI,
"La synthèse d'image", Editions HERMES, janvier 1988, pp. 103-105.

[PITT 80] : M.L.V. PITTEWAY and D.J. WATKINSON,
"Bresenham's Algorithm with Grey Scale", Communications of the ACM, vol. 23, No. 11, Nov. 1980, pp. 625-626.

[TURK 82] : K. TURKOWSKI,
"Anti-Aliasing through the Use of Coordinate Transformations", ACM Transactions on Graphics, vol. 1, No. 3, July 1982, pp. 215-234.

[WARN 69] : J. WARNOCK,
"A Hidden-Surface Algorithm for Computer Generated Half-Tone Pictures", University of Utah, TR 4-15, 1969, NTIS AD-753 671.

[WATK 70] : G. WATKINS,
"A Real-Time Visible Surface Algorithm", University of Utah, UTEC-CSc-70-101, June 1970, NTIS AD-76 004.

[WHIT 80] : T. WHITTED,
"An Improved Illumination Model for Shaded Display", Communication of the ACM, vol. 23, No. 6, June 1980, pp. 343-349.

[WHIT 83] : T. WHITTED,
"Anti-Aliased Line Drawing Using Brush Extrusion", Computer Graphics, vol. 17, NO. 3, July 1983, pp. 151-156.

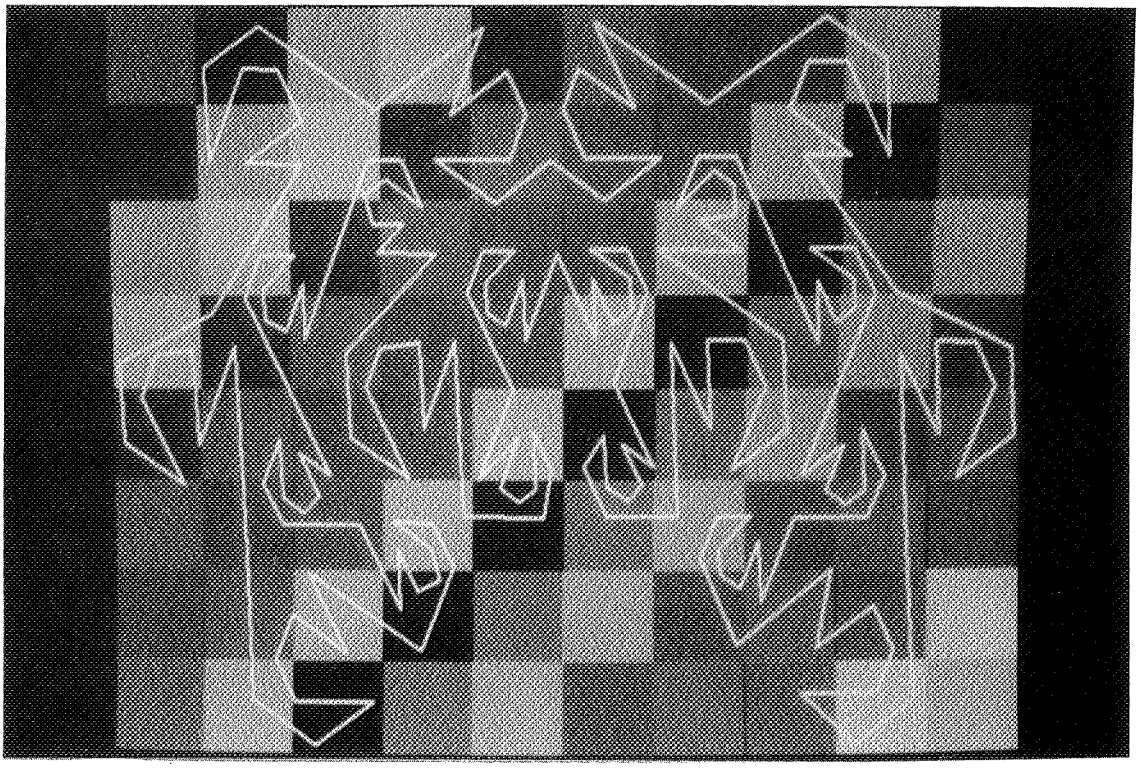


Image III.1 : Le phénomène de marches d'escalier sur les segments.

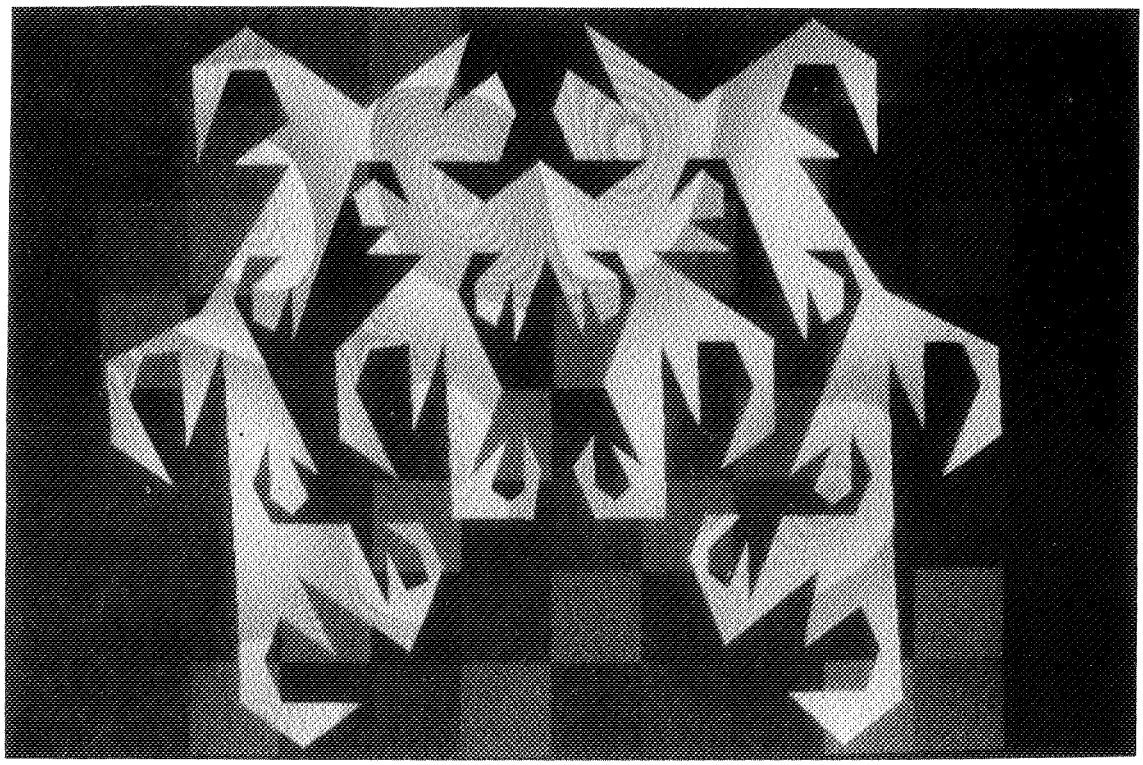


Image III.2 : Le phénomène de marches d'escalier sur les bords d'un polygone.

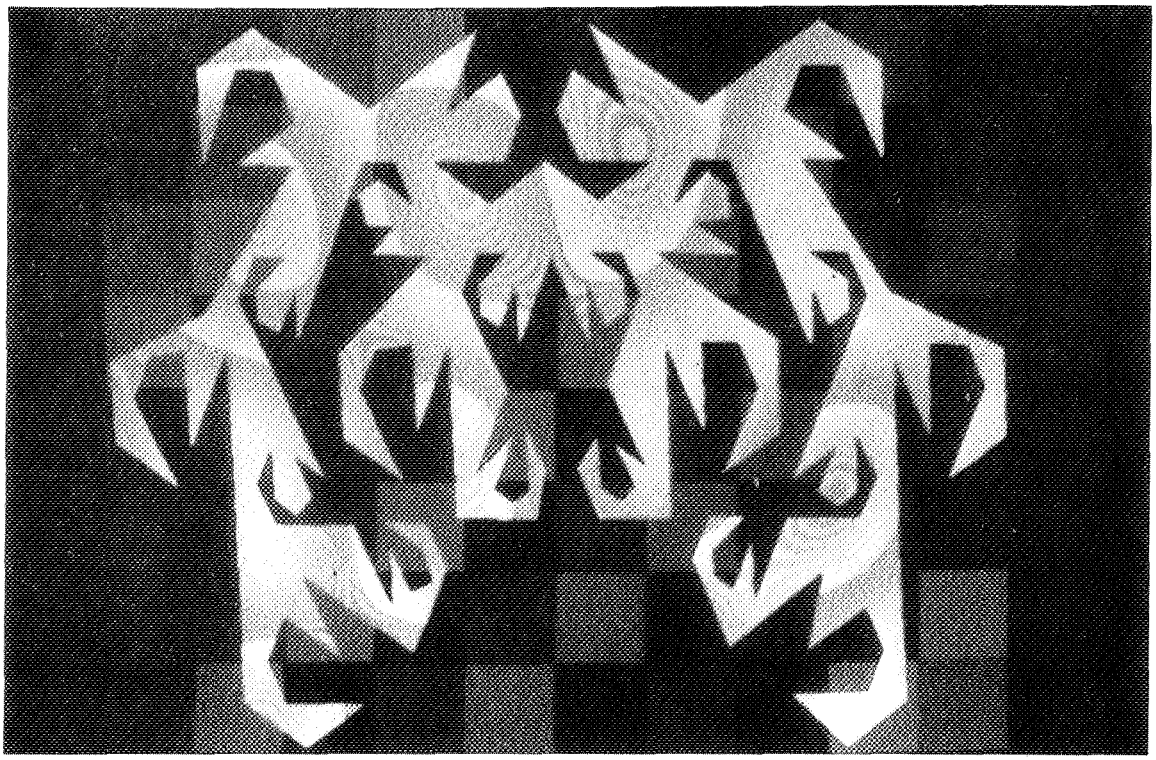


Image III.3 : Méthode de post-filtrage en utilisant une fenêtre de Hamming.

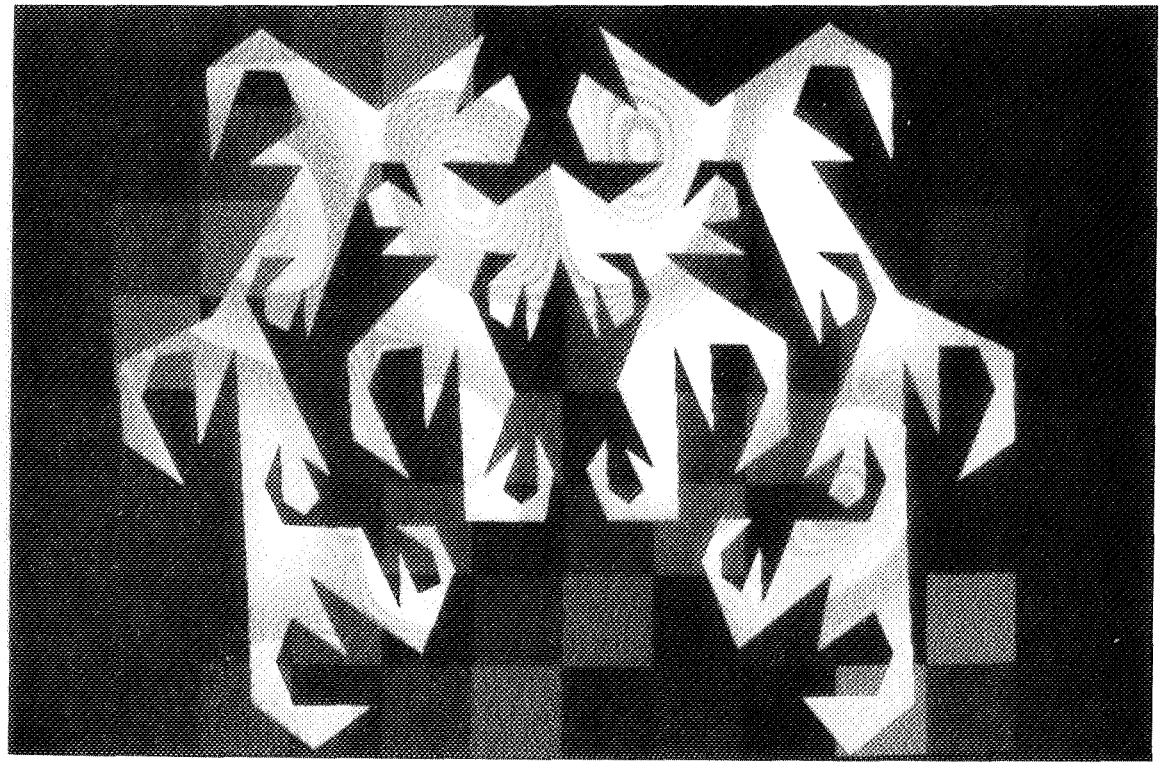


Image III.4.a : Méthode de sur-échantillonnage, x_sur= 1 et y_sur=5.

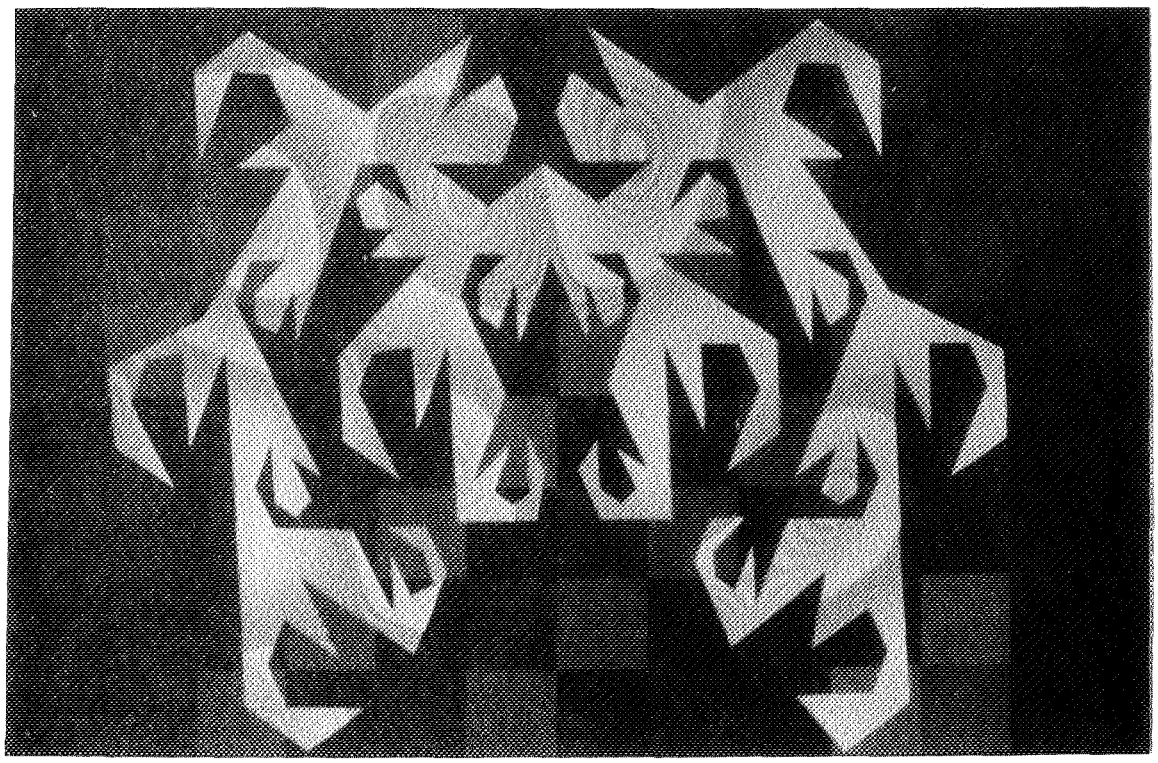


Image III.4.b : Méthode de sur-échantillonnage, x_sur= 5 et y_sur=5.

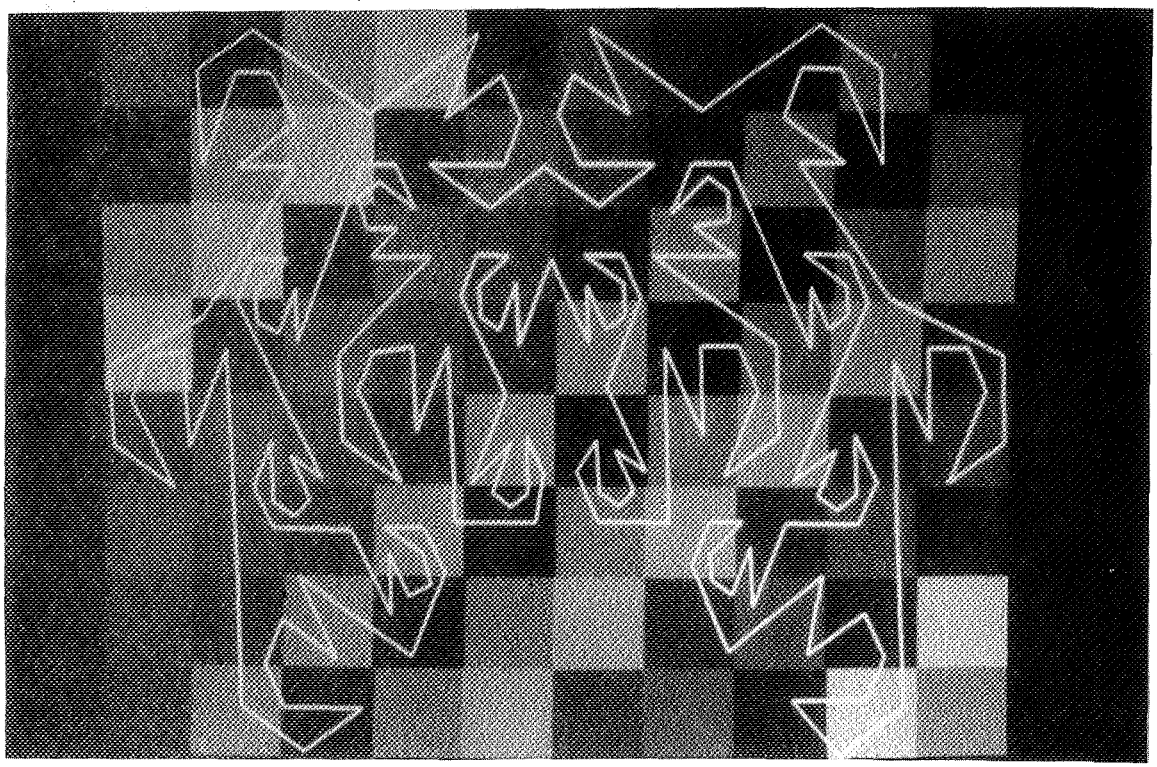


Image III.5 : Méthode de sur-échantillonnage pour des segments d'épaisseur d'un pixel, x_sur= y_sur=5.

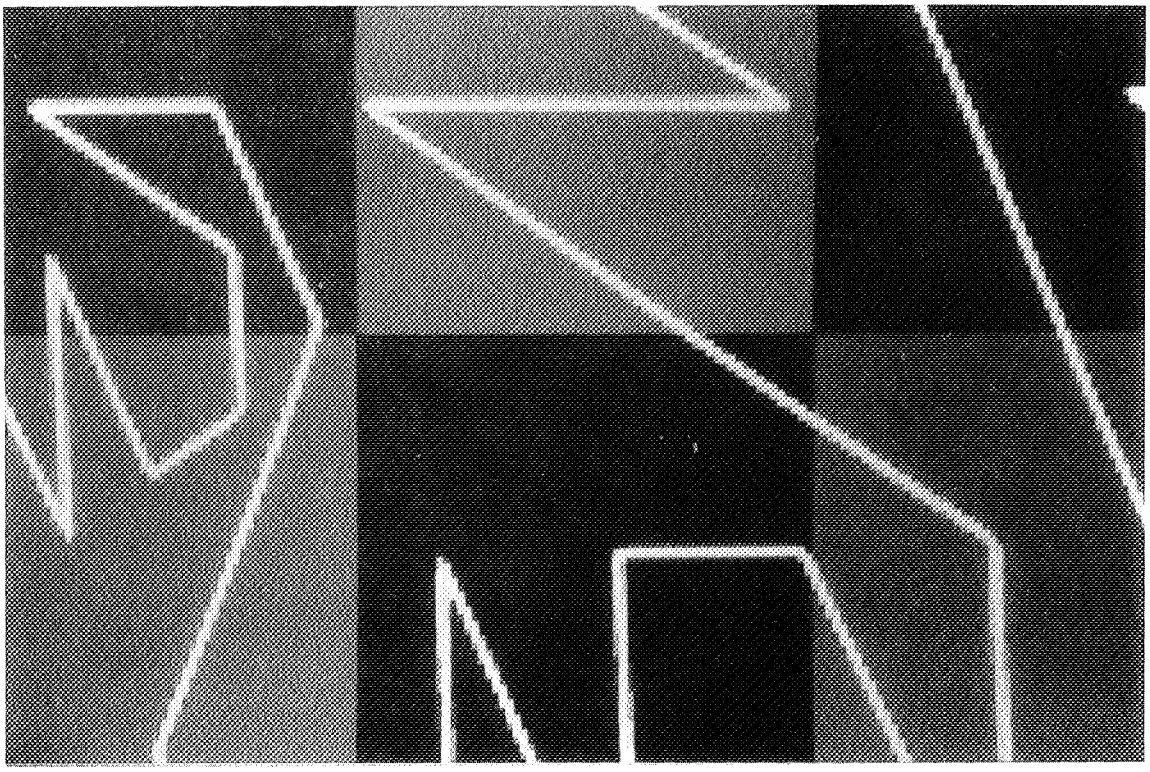


Image III.6 : Zoom x5 de l'image III.5.

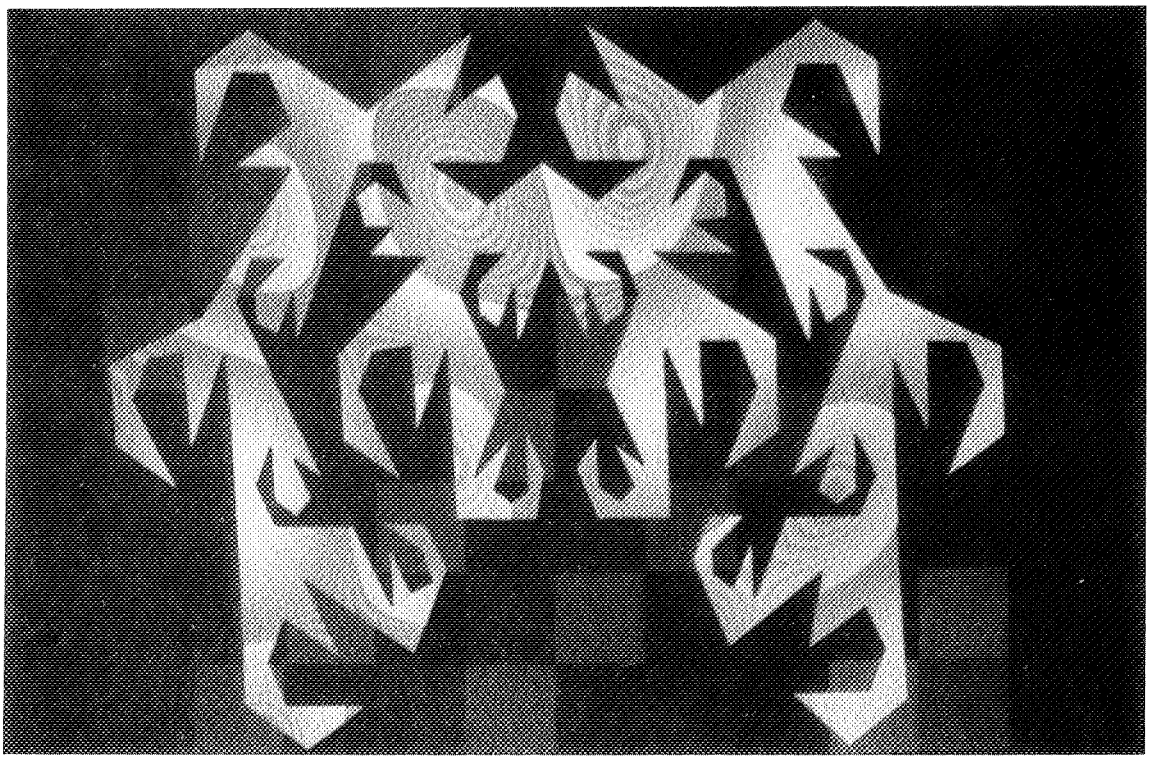


Image III.7 : Résultats obtenus avec la modification (deux pixels traités par ligne ou par colonne selon le cas) de la méthode de [PIWA 80].

CHAPITRE IV

Antialiassage et tampon de profondeur

CHAPITRE IV

Antialiassage et tampon de profondeur

IV.1 Introduction

Dans le chapitre précédent, nous avons étudié le problème de l'antialiassage des objets géométriques, d'abord dans le cas 2D, puis pour quatre techniques d'élimination des parties cachées parmi les plus importantes. Etant donné que le tampon de profondeur ("z-buffer"), introduit en 1974 par Catmull [CATM 74], est la technique la plus simple et peut être la plus répandue d'élimination des parties cachées à l'heure actuelle, les problèmes d'antialiassage liés à cette technique font l'objet d'une étude plus détaillée dans ce chapitre. Comme nous l'avons remarqué au chapitre précédent, pour des raisons de simplicité, les scènes 3D sont souvent facettisées par des algorithmes tels que [LANE 80]. L'antialiassage des scènes polygonales visualisés à l'aide d'un tampon de profondeur sera donc étudié dans ce chapitre.

Après un bref rappel des principes de base du tampon de profondeur, nous exposerons les problèmes spécifiques rencontrés dans l'application d'un traitement d'antialiassage associé à cette technique. Puis, nous décrirons plusieurs méthodes résolvant les problèmes d'antialiassage pour un affichage utilisant un tampon de profondeur. Enfin, nous proposerons deux méthodes originales d'antialiassage des scènes polygonales qui préservent la simplicité du tampon de profondeur, dont une méthode intéressante qui réserve les calculs les plus coûteux (antialiassage, texturage, lissage, ...) uniquement pour les pixels visibles de la scène.

IV.2 Principes du tampon de profondeur

La méthode du tampon de profondeur pour l'élimination des parties cachées d'une scène 3D (en général polygonale) est une des méthodes les plus utilisées. De par de sa simplicité, de nombreux systèmes graphiques disposent d'une version cablée (ou microprogrammée) de cet algorithme qui réalise un affichage très rapide d'une scène 3D.

L'image produite est représentée sous forme matricielle ; en plus de la couleur, la profondeur au centre de chaque pixel est disponible. Les profondeurs sont initialisées à la valeur maximale avant l'affichage de la première entité. L'élimination des parties cachées est réalisée en comparant la profondeur de chaque pixel (x,y) d'une entité à afficher à la profondeur courante de ce pixel mémorisée précédemment. Si cette profondeur est inférieure à la profondeur courante, on met la couleur et la profondeur de l'entité à afficher en ce pixel à la place des valeurs courantes. Les principes de cette méthode sont décrits par l'algorithme suivant .

Algorithme IV.1, le tampon de profondeur :

```
{
 pour chaque pixel (x,y) de l'écran faire z(x,y):=z_max;   /* z(x,y) : la profondeur du pixel (x,y) */
 pour chaque entité E de la scène faire
  pour chaque pixel (x,y) où E se projette faire
   {
    z:= profondeur de E en (x,y);
    si z<z(x,y) alors
     {
      z(x,y):=z;
      c(x,y):= couleur de E en (x,y);     /* c(x,y) : la couleur du pixel (x,y) */
     }
   }
}
```

Pour la suite de ce chapitre, il est important de remarquer les propriétés suivantes du tampon de profondeur :

- les entités composant la scène peuvent être affichés dans un ordre quelconque, sans aucun tri préalable ;
- les intersections entre les entités de la scène ne constituent pas un cas particulier et elles sont prises en compte tout naturellement ;
- l'image produite est en général connue à la définition de l'affichage.

IV.3 Position du problème

Le problème rencontré pour effectuer un traitement d'antialiassage avec la méthode du tri en profondeur dans le cas des polygones adjacents (cf. chapitre III) se retrouve pour cette méthode. De plus, la simplicité du tampon de profondeur est à l'origine d'obstacles supplémentaires pour un antialiassage correct. Comme les polygones peuvent être affichés dans un ordre quelconque et comme l'affichage des polygones intersectants ne nécessite pas de considérations particulières, contrairement à la méthode de tri en profondeur, deux nouveaux problèmes d'antialiassage apparaissent : le changement de couleur de fond et

l'intersection entre des polygones. Pour mettre ces problèmes en évidence, nous utilisons une scène polygonale simple (cf. image IV.1).

IV.3.1 Changement de couleur de fond

Avec la méthode du tampon de profondeur, les polygones composant la scène 3D peuvent être affichés dans un ordre quelconque sans aucun tri préalable et en particulier sans tri en profondeur. Ceci peut provoquer un changement de couleur de fond, problème que nous allons étudier dans ce qui suit.

Le polygone courant p_j est partiellement caché par le polygone p_i qui est déjà affiché (cf. figure IV.1). Chaque pixel (x,y) du bord AB appartenant à p_i a été traité, lors de l'affichage de p_i, avec les couleurs (ou intensités) correspondant à p_i et au fond en (x,y). L'affichage de p_j change le fond et par conséquent les couleurs calculées précédemment pour tous les pixels de AB ne sont plus correctes (cf. image IV.2). Ce défaut devient très visible quand la couleur de p_j et celle du fond contrastent fortement. Lors de l'affichage de p_j, il n'existe plus aucune information ni sur la couleur initiale de p_i en (x,y), ni sur la répartition du pixel (x,y) entre p_i et p_j. Or ces deux informations sont indispensables pour effectuer un antialiassage correct. De plus, la détection des pixels de AB n'est pas évidente à cet instant. Le changement de couleur de fond est donc un des problèmes les plus importants à résoudre pour l'antialiassage des images obtenues en utilisant un tampon de profondeur.

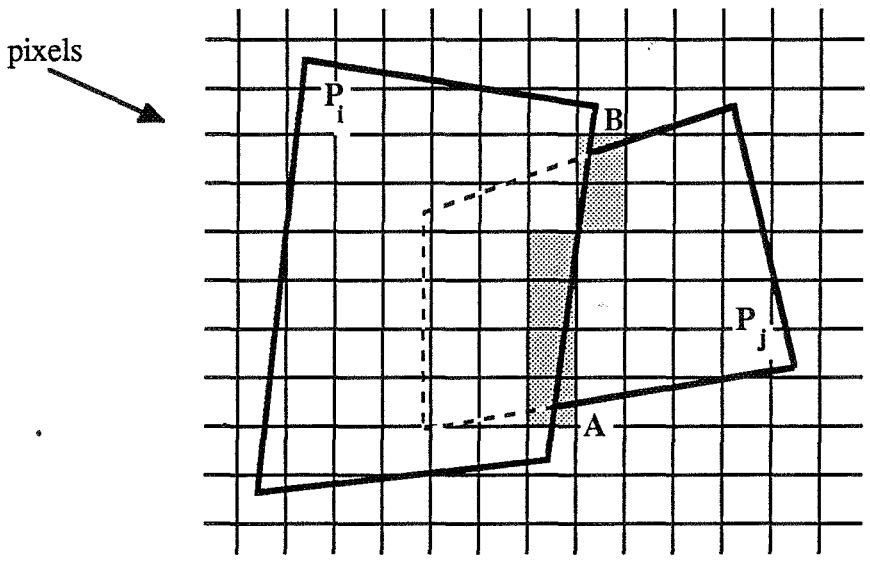


Figure IV.1 : Problème de changement de couleur de fond pour AB appartenant à p_i (affiché avant p_j).

IV.3.2 Intersection des polygones

Le polygone courant p_j intersecte un polygone déjà affiché p_i. Un nouveau bord est créé entre ces deux polygones le long de leur intersection AB (cf. figure IV.2). Comme AB n'est un bord explicite ni de p_j, ni de p_i, la détection des pixels de ce nouveau bord et leur antialiassage posent problème.

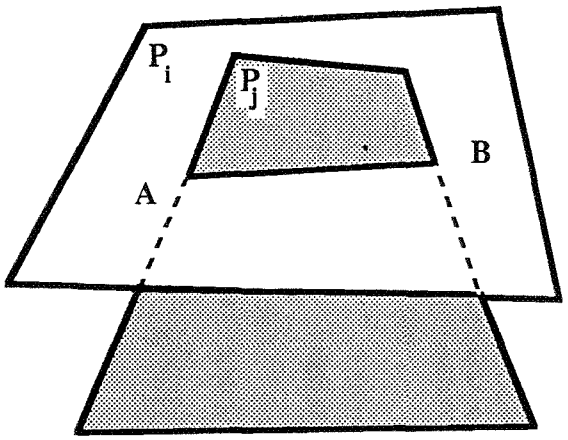


Figure IV.2 : Problème de l'intersection AB entre deux polygones.

IV.4 Méthodes d'antialiassage

Les trois problèmes (polygones adjacents, changement de couleur de fond et intersection des polygones) montrent qu'un antialiassage exact du tampon de profondeur est difficilement envisageable, à moins d'utiliser soit une solution plutôt théorique qui gère la liste de tous les polygones intersectants chaque pixel, soit la solution du sur-échantillonnage global. Avant d'étudier quelques méthodes connues ou nouvelles, nous proposons une méthode de sur-échantillonnage global pour le tampon de profondeur.

IV.4.1 Méthode de sur-échantillonnage global en plusieurs étapes

On a vu dans le chapitre III que le sur-échantillonnage global est la solution générale au problème de l'aliassage dans la plupart des cas. Comme nous l'avons remarqué, cette solution est coûteuse en temps de calcul et elle est, en général, gourmande en espace mémoire (cf. chapitre III). Etant donnés les nombreux calculs de profondeur et leur

comparaison (pour les sous-pixels), il est intéressant de décharger le processeur central de ce type de calcul. De plus, il faut diminuer l'occupation en espace mémoire. Pour ces deux raisons, nous proposons une méthode de sur-échantillonnage global par étapes successives qui utilise la possibilité d'un affichage rapide, en particulier dans le cas d'un tampon de profondeur cablé.

Si les rapports de sur-échantillonnage sont m et n suivant respectivement x et y, la scène 3D initiale peut être divisée en m x n sous-scènes 3D. Chaque sous-scène s'affiche à la définition d'affichage en utilisant un affichage rapide, le résultat est une image I. Toute image I sera filtrée à l'aide d'un filtre passe-bas et rééchantillonnée à la définition de $\frac{1}{m} \times \frac{1}{n}$ de celle d'affichage pour obtenir une sous-image correspondant à une sous-scène. L'encombrement de la mémoire centrale est donc à peu près celui d'une image à la définition d'affichage. Les temps de calculs nécessaires sont essentiellement ceux du filtrage d'une image de définition m x n.

A part le problème du découpage 3D de la scène initiale en sous-scènes représentables à la définition d'affichage, la difficulté de cette méthode est le filtrage au voisinage des bords de sous-scènes et la continuité de l'image finale sur la frontière entre deux sous-images produites. Pour résoudre ce problème, les sous-scènes peuvent se chevaucher de p pixels si la taille du support du filtre utilisé (cf. chapitre II) est 2p ou 2p+1 pixels. Cette méthode est simple et générale. Elle peut résoudre entièrement les trois problèmes cités ci-dessus pour des scènes quelconques, pas forcément polygonales. De plus, comme tout autre sur-échantillonnage global, les problèmes de petits objets ou des sommets de polygones sont résolus. L'inconvénient majeur ce cette méthode est naturellement les temps de calculs pour le filtrage d'une image haute définition.

IV.4.2 Rappel de quelques méthodes connues

Les publications au sujet de l'antialiassage des images affichées par un tampons de profondeur sont peu nombreuses, peut être pour des raisons commerciales ou à cause des difficultés d'avoir une méthode à la fois simple et générale. On peut cependant citer les articles suivants :

En 1984, Carpenter [CARP 84] a publié une méthode appelée A-buffer. Cette méthode est intéressante pour obtenir des images de bonne qualité mais comme celle de Catmull [CATM 78], elle est assez complexe et coûteuse. Dans cette méthode, pour chaque pixel, un mot est utilisé pour coder l'information de profondeur et un deuxième mot est utilisé pour mémoriser une couleur ou un pointeur sur un enregistrement (contenant, entre autre, des

informations de type géométrique) d'une taille plus importante. Par conséquent, pour chaque pixel, plusieurs mots peuvent être nécessaires. Le A-buffer nécessite la disponibilité d'une mémoire de taille assez importante : l'auteur préconise l'utilisation d'une mémoire virtuelle. Cette méthode donne de bons résultats graphiques (antialiassage et transparence), mais n'a ni la simplicité ni l'autonomie du tampon de profondeur, à cause de la manipulation d'une grande quantité d'informations pour chaque pixel et par conséquent l'utilisation d'une mémoire virtuelle. De plus cette méthode est assez lente.

Evans a présenté en 1984 un résumé de sa méthode dans [EVAN 84]. Avec quelques approximations, cette méthode antialiasse les contours apparents des objets composés de polygones ombrés avec un lissage du type Gouraud. Pour chaque pixel de l'écran, un bit est utilisé pour marquer les pixels antialiassés afin de distinguer les pixels de contours apparents des pixels appartenant au bord de deux polygones adjacents. Cependant, dans le cas général, le changement de couleur de fond, le problème des polygones adjacents non-ombrés et le bord crée par l'intersection entre deux polygones ne peuvent pas être résolus par cette méthode.

Fujimoto, Perrott et Iwata [FUJI 84] ont proposé une méthode en 1984. Les auteurs présentent un système graphique 3D avec un tampon de profondeur et antialiassage. La méthode de traitement est approximative et pas très détaillée dans l'article. Tous les problèmes ne sont pas résolus par cette méthode ; en fait, le cas de changement de la couleur du fond et l'intersection de deux polygones ne sont pas pris en compte. D'ailleurs, les auteurs proposent le cas de changement de la couleur du fond comme un sujet de futures recherches.

En 1985, Duff [DUFF 85] a publié sa méthode pour un antialiassage avec un affichage utilisant la technique du tampon de profondeur. Dans cette méthode, qui est une suite de [PORT 84], il mémorise les profondeurs courantes aux quatre coins des pixels et il les compare aux profondeurs de chaque nouvel objet aux mêmes points . Pour chaque pixel, si après comparaison, les quatre coins ont le même signe, le pixel est uniforme. Dans le cas contraire, le pixel est partagé entre le fond courant et l'objet à afficher. Suivant les signes des quatre coins d'un pixel, on distingue seize cas possibles : 2 cas de pixel uniforme et 14 cas de pixel partagé peuvent exister. L'information sur la répartition d'un pixel entre le fond et un nouvel objet de la scène est obtenue par des interpolations linéaires sur les profondeurs de l'objet et les profondeurs courantes aux quatre coins du pixel afin de trouver les valeurs de profondeurs égales sur les cotés du pixel. Cette opération indique approximativement les deux points où l'objet intersecte les deux côtés du pixel. L'aire de la surface couverte par le

fond ou l'objet est donc calculée ("area sampling"), ce qui correspond au calcul d'une moyenne simple, comme beaucoup d'autre méthodes.

La méthode de Duff et relativement simple et elle peut traiter le cas d'intersection entre des polygones. L'auteur précise que cette méthode comporte des approximations et même dans certains cas, de petites erreurs. Il signale d'ailleurs que cette méthode ne peut traiter ni les petits objets de la scène, ce qui est également le cas de la plupart des autres méthodes, ni le cas des polygones adjacents. De plus nous pouvons remarquer que cette méthode ne peut pas résoudre le cas du changement de couleur de fond.

La méthode du zz-buffer présentée en 1989 par Salesin et Stolfi [SALE 89] est une variante sophistiquée du tampon de profondeur. Il y a bien utilisation d'un tableau, appelé zz, mais une case de ce tableau pointe sur un enregistrement correspondant à une cellule rectangulaire de taille donnée de l'écran. Un enregistrement contient, entre autres, un pointeur sur une liste d'objets susceptibles d'être visibles dans la cellule et sur deux nombres donnant la profondeur minimale et maximale des objets contenus dans la cellule. Comme on le voit, le zz-buffer admet une certaine ressemblance avec le A-buffer. Pour résoudre les problèmes d'aliassage, le zz-buffer procède ainsi : pour chaque pixel de la cellule, une procédure tente de déterminer la couleur à afficher. Si la situation est trop compliquée, elle remplace le calcul au centre du pixel par un sur-échantillonnage stochastique (cf. chapitres I et II). Comme le A-buffer, cette méthode est assez lente.

IV.4.3 La méthode du tampon en g et en z

IV.4.3.1 Introduction

On a vu précédemment que l'antialiassage associé à l'algorithme du tampon de profondeur soulève de nombreux problèmes qui rendent très difficile un traitement précis. Nous proposons, maintenant, une méthode appelée tampon en g et en z ou gz-buffer [GHAZ 85]. Cette méthode d'antialiassage 3D peut traiter les trois problèmes signalés ci-dessus pour une scène polygonale 3D. Cependant, il subsiste des approximations inévitables et quelques cas particuliers.

IV.4.3.2 Description de la méthode

Dans la méthode du gz-buffer, nous avons essayé de garder la simplicité et l'autonomie offertes par le tampon de profondeur. Le gz-buffer (cf. figure IV.3) utilise comme espace

mémoire un tampon de profondeur conventionnel ainsi qu'un tampon géométrique (g-buffer) de taille moins importante qui mémorise une information de type géométrique g pour chaque pixel (x,y). Cette information géométrique est une fonction de la surface du pixel couverte par le dernier polygone qui intersecte le pixel. La valeur g de chaque pixel indique si celui-ci a déjà subi un traitement d'antialiassage ou non. Supposons que g soit codé sur les n bits du g-buffer (par exemple, pour une mémoire d'image sur laquelle chacune des primitives RVB est codée sur un octet, il semble raisonnable de consacrer quatre bits par pixel au g-buffer). L'information géométrique est donc représentée par une valeur entière g avec n bits de précision, $0 \leq g \leq 2^n-1$. Au début, la valeur g de chaque pixel est initialisée à 2^n-1. Si $g=2^n-1$, le pixel est entièrement couvert par le dernier polygone qui l'a intersecté (le fond initial est considéré comme le premier polygone), il n'est donc pas partagé ; $g \neq 2^n-1$ signifie donc qu'un pixel a déjà été antialiassé.

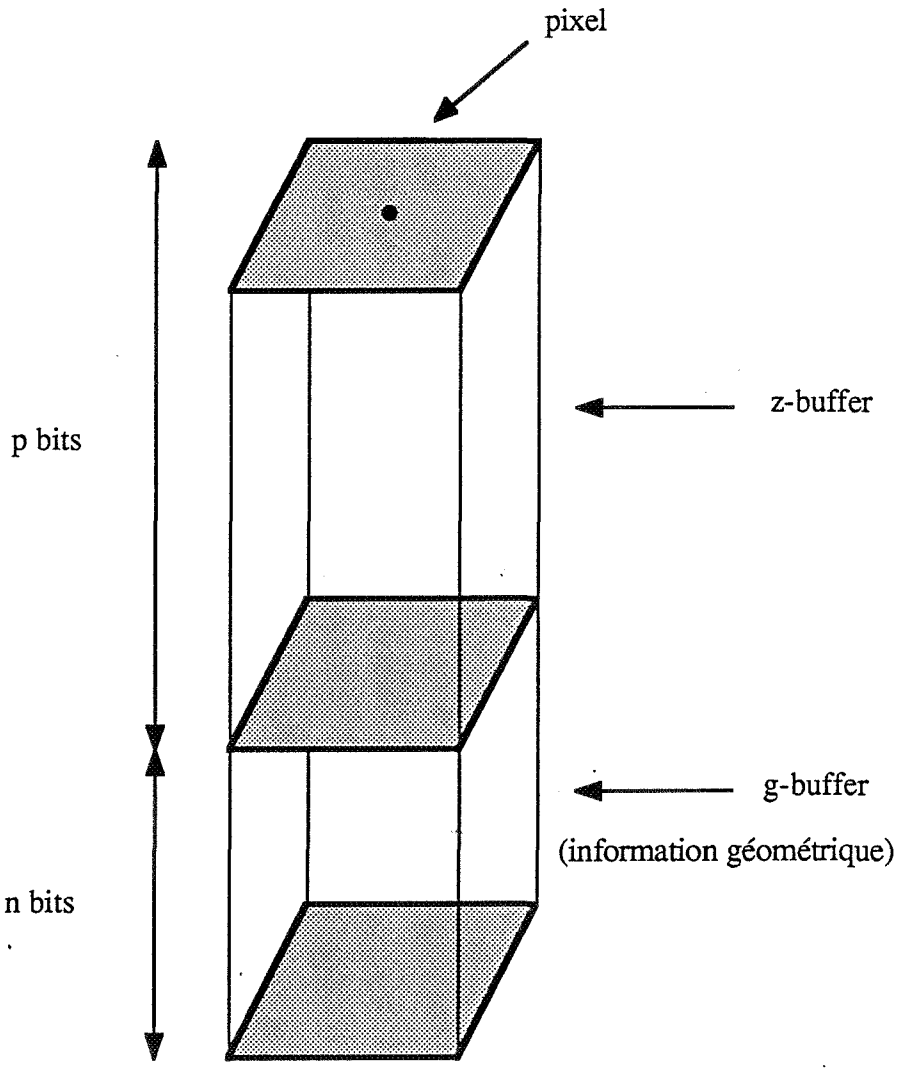


Figure IV.3 : Structure du gz-buffer.

Le traitement d'antialiassage utilisé pour les deux premières versions du gz-buffer [GHAZ 85] et [BEIG 86] est le sur-échantillonnage. Comme nous l'avions signalé dans la première version du gz-buffer [GHAZ 85], il est intéressant d'utiliser une méthode d'antialiassage rapide au lieu du sur-échantillonnage. La nouvelle version du gz-buffer [GHAZ 87] que nous présentons ici, utilise la méthode de Pitteway et Watkinson [PITT 80] pour antialiasser les pixels de bords et les intersections entre les polygones. Le choix de cette méthode est justifié d'une part par sa simplicité (cf. chapitre III), d'autre part par le fait qu'un seul pixel par ligne ou par colonne, selon la pente du bord, est antialiassé. Le gz-buffer exploite cette particularité, comme nous le verrons plus loin.

Considérons le traitement d'un polygone p_j. p_j est d'abord affiché par un tampon de profondeur avec une couleur réservée r, qui est hors du spectre des couleurs utilisées dans la scène. En effet, l'affichage de p_j avec la couleur r fournit un masque 3D, ce qui évite des calculs de profondeur et des comparaisons nécessaires pour des tests de visibilité qui augmentent les temps de calcul dans [GHAZ 85] et [BEIG 86]. L'utilisation d'un masque est particulièrement avantageux si un tampon de profondeur cablé ou même microprogrammé est disponible pour l'affichage.

Après l'affichage du masque, la boîte englobante de p_j est balayée. Soit c(x,y), la couleur du pixel (x,y). Si c(x,y)=r, alors le pixel (x,y) appartient à p_j et on affectera la valeur de 2^n-1 à g(x,y). Si c(x,y)$\neq$ r, g(x,y)$\neq 2^n$-1 et si au moins une des couleurs des quatre pixels voisins : (x-1,y), (x+1,y), (x,y-1) et (x,y+1) est égale à r, alors c'est un cas de changement de couleur de fond (cf. figure IV.1). Dans ce cas, le pixel (x,y) antialiassé appartient à un polygone déjà affiché p_i et aucune information exacte n'est disponible sur la couleur de p_i au point (x,y), ou sur la répartition du pixel (x,y) entre p_i et le fond. Cette couleur est approximée par celle d'un pixel voisin approprié. L'erreur est nulle quand p_i est un polygone de couleur uniforme ; elle est faible quand p_i est un polygone texturé car la texture doit être déjà antialiassée et les pixels contigus recouverts par celle-ci ont donc des couleurs très proches grâce au filtrage passe-bas que la texture a subi (cf. chapitre V).

Une fois qu'un pixel (x,y) concerné par un changement de couleurs de fond est détecté, le problème restant est celui du choix du pixel voisin approprié. L'algorithme du gz-buffer détermine d'abord si le pixel (x,y) appartient à un bord plutôt horizontal ou plutôt vertical. Ceci peut être réalisé en testant deux des pixels voisins directs : (x-1,y) et (x+1,y). Si g(x-1,y)=g(x+1,y)=2^n-1 et si c(x-1,y)$\neq$ r ou c(x+1,y)$\neq$ r, alors le bord est considéré plutôt vertical (cf. figure IV.4), sinon il est considéré plutôt horizontal. Ce test détecte correctement l'orientation d'un bord car, comme nous le verrons plus tard, pour les bords de chaque polygone déjà affiché, un seul pixel par ligne ou par colonne, selon la pente du bord,

est antialiassé grâce à l'emploi de l'algorithme de Pitteway et Watkinson. Dans le cas d'un bord plutôt vertical, la couleur choisie pour p_i en (x,y) est celle du pixel (x-1,y) si c(x-1,y)≠r, celle du pixel (x+1,y) sinon. De même, pour un bord plutôt horizontal, c'est la couleur d'un des deux pixels (x,y-1) ou (x,y+1) qui est utilisée.

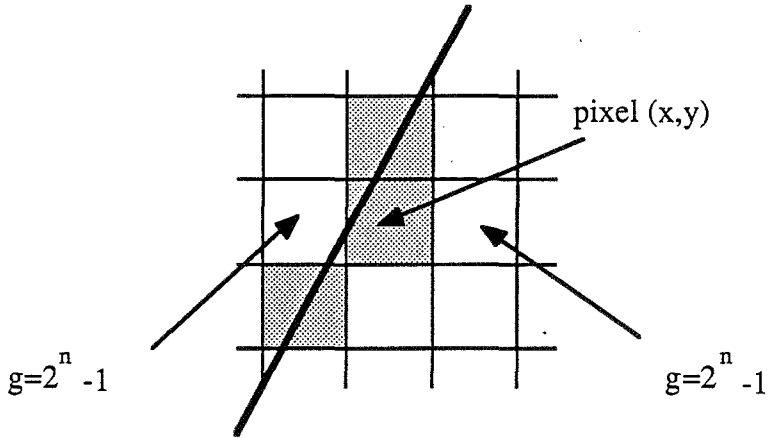


Figure IV.4 : Détection de l'orientation d'un bord.

Soit c_i la couleur lue, et $c_j(x,y)$ la couleur de p_j en (x,y). L'information géométrique étant la valeur g(x,y), la nouvelle couleur du pixel (x,y), c(x,y), est définie par :

$$c(x,y) = c_i.g(x,y)/(2^n-1)+c_j(x,y).[1-g(x,y)/(2^n-1)]$$
$$= [c_i-c_j(x,y)].g(x,y)/(2^n-1)+c_j(x,y)$$

Après avoir balayé p_j, les bords visibles de p_j sont antialiassés en utilisant l'algorithme de Pitteway et Watkinson, qui fournit une information géométrique a(x,y) pour chaque pixel (x,y) de bord (cf. chapitre III). On sait que $0 \leq a(x,y) \leq 1$ représente l'aire de la surface couverte par le dernier polygone affiché sur ce pixel de bord. La valeur a(x,y) est codée sur les n bits du g-buffer. Nous obtenons ainsi une nouvelle valeur de g(x,y) pour un pixel antialiassé. Pour traiter chaque pixel (x,y), on devrait utiliser les couleurs de p_j et du fond en (x,y). Rappelons que le fond est soit le fond d'origine, soit un polygone déjà affiché. Pour éviter le problème des bords adjacents (cf. chapitre III), la couleur du fond en ce pixel est approximée par celle du pixel voisin approprié. Comme dans le cas précédent, l'erreur due à ce choix est nulle ou négligeable. Soient f la couleur approximée du fond en (x,y) et $c_j(x,y)$ la couleur de p_j au pixel (x,y) ; la couleur du pixel (x,y) devient :

$$c(x,y) = c_j(x,y).a(x,y)+f.[1-a(x,y)]$$
$$= a(x,y).[c_j(x,y)-f]+f$$

Une fois les pixels des bords de p_j antialiassés, les éventuels bords créés par des intersections (cf. figure IV.2) sont détectés et antialiassés. La méthode proposée détecte les points extrémités des intersections. Cela peut être fait dans la plupart des cas en testant les valeur de g. Rappelons que pour un pixel antialiassé, $g \neq 2^n-1$. On distingue un point extrémité d'une arête intersection d'un sommet d'un polygone par le nombre des pixels voisins antialiassés. Comparons un sommet de polygone (cf. figure IV.5.a), et un point extrémité d'une arête d'intersection (cf. figure IV.5.b) : dans le premier cas, deux des huit pixels voisins sont antialiassés alors que, dans le second cas, uniquement un des huit pixels voisins l'est. Les extrémités de l'intersection peuvent être ainsi déterminés et le bord créé par l'intersection est antialiassé par l'algorithme de Pitteway et Watkinson.

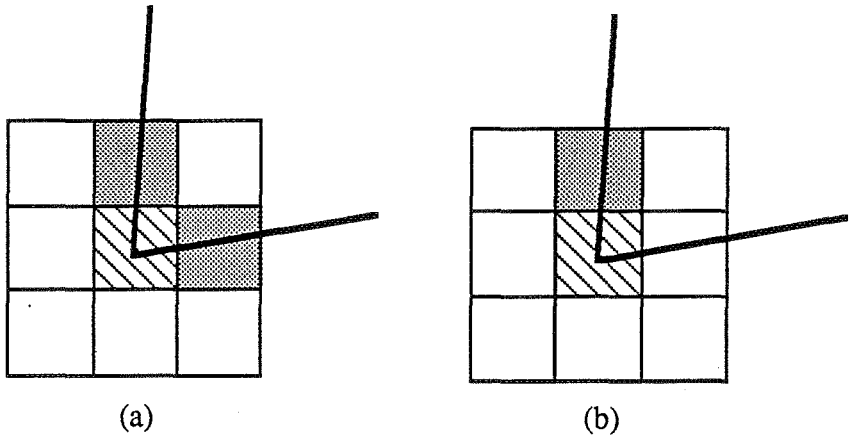


Figure IV.5 : La différence entre un sommet de polygone (a) et un point extrémité d'une intersection (b) pour l'algorithme du gz-buffer.

Pour la dernière étape, on balaie le masque p_j une fois de plus. Pour chaque pixel (x,y), si c(x,y)=r, alors (x,y) est un pixel visible à l'intérieur de p_j et il est affiché avec la couleur de la texture de p_j au pixel (x,y) : $c(x,y) = c_j(x,y)$.

Les principes de l'algorithme du gz-buffer sont résumés par le pseudo code suivant .

Algorithme IV.2, le gz-buffer :

```
{
pour tous les pixels (x,y) de l'écran faire
  g(x,y):=2^n-1;     /* initialisation du g-buffer */
pour chaque polygone courant p_j faire
{
  afficher p_j avec une valeur constante r; /* r est hors du spectre des couleurs utilisées dans la scène */

  pour chaque pixel (x,y) de la boîte englobante de p_j faire
   si c(x,y)=r alors g(x,y):=2^n-1
   sinon
    si g(x,y)≠ 2^n-1 et (c(x-1,y)=r ou c(x+1,y)=r ou c(x,y-1)=r ou c(x,y+1)=r) alors
     {  /* c'est un cas de changement de couleur de fond */
       si g(x-1,y)=g(x+1,y)=2^n-1 et (c(x-1,y)≠ r ou c(x+1,y)≠ r) alors
        /* c'est un bord plutôt vertical */
         si c(x+1,y)≠ r alors c_i:=c(x+1,y)
         sinon c_i:=c(x-1,y)   /* c_i est considéré comme la couleur de p_i en (x,y) */
       sinon    /* c'est un bord plutôt horizontal */
         si c(x,y+1)≠ r alors c_i:=c(x,y+1)
         sinon c_i:=c(x,y-1) ;
       c(x,y):=[c_i-c_j(x,y)].g(x,y)/(2^n-1)+c_j(x,y);   /* c_j(x,y) est la couleur de p_j en (x,y) */
     }

  déterminer la couleur du fond f et antialiasser les pixels visibles des bords de p_j :
    c(x,y):=a(x,y).[c_j(x,y)-f]+f;
  écrire la nouvelle valeur de g(x,y);
  détecter et traiter les éventuels bords créés par des intersections entre p_j et d'autres
  polygones déjà affichés;
  pour chaque pixel (x,y) de la boîte englobante de p_j faire
   si c(x,y)=r alors c(x,y):=c_j(x,y) ;
}
}
```

IV.4.3.3 Conclusion

Dans la plupart des cas, cette méthode est efficace et elle donne de bons résultats graphiques malgré quelques approximations, la contrainte de n'antialiasser qu'un seul pixel par ligne ou par colonne et l'utilisation d'une moyenne simple. Les résultats obtenus par cette méthode, pour une scène 3D présentant les trois types de problèmes associés à la technique du tampon de profondeur, sont visualisés sur l'image IV.3. L'image IV.1 est l'affichage par un tampon de profondeur ordinaire et l'image IV.2 est le résultat de l'antialiassage 2D. L'image IV.5 représente le résultat d'un traitement avec cette méthode pour une scène texturées.

Cette méthode est assez rapide et elle conserve tous les avantages du tampon de profondeur : la simplicité et l'autonomie, en particulier, l'inutilité des tris. Les suppléments

en mémoire et en temps de calcul sont raisonnables. A part les comparaisons, cette méthode n'est pas beaucoup plus coûteuse en temps "C.P.U." que la méthode 2D de Pitteway et Watkinson. Pour la scène représentée par l'image IV.1, le temps "C.P.U." nécessaire pour cette méthode est 10 fois moindre que pour la version du gz-buffer [BEIG 86] utilisant un sur-échantillonnage. Il subsiste cependant quelques défauts :

- comme l'algorithme de Pitteway et Watkinson, cette méthode ne peut pas traiter exactement les sommets des polygones, les pixels qui sont couverts par plus de deux polygones et les petits objets, ce qui est le cas de la plupart des méthodes autres que le sur-échantillonnage global ;
- dans le cas de deux bords plutôt verticaux, presque parallèles et distants d'environ un pixel (un objet localement petit), le test sur g(x-1,y) et g(x+1,y) ne peut déterminer l'orientation du bord, c'est également le cas de [GHAZ 85] et de [BEIG 86] ;
- quand un point extrémité d'une intersection appartient à un bord visible déjà traité ou même s'il est le voisin direct de ce bord, le test sur les huit pixels voisins n'est pas suffisant pour le détecter ; ce n'est pas le cas de [GHAZ 85] et de [BEIG 86] qui utilisent d'autre méthodes pour la détection des pixel d'intersection.

L'algorithme du gz-buffer présenté dans ce chapitre peut être avantageusement utilisé avec un tampon de profondeur cablé ou microprogrammé. De plus, grâce à sa simplicité, son implantation matérielle (ou en microprogrammation) est envisageable. Comparé à un tampon de profondeur cablé, uniquement un nombre restreint de plans mémoire (4 plans par exemple) pour le g-buffer est nécessaire (cf. figure IV.3). Les principales opérations utilisées sont classiques : lecture, écriture et comparaison de profondeurs des pixels. En ce qui concerne la méthode d'antialiassage utilisée, à notre connaissance, la méthode de Pitteway et Watkinson est la plus simple à implanter et d'ailleurs la plus répandue dans les systèmes graphiques disposant d'un antialiassage 2D.

IV.4.4 Antialiassage hybride par étapes successives

IV.4.4.1 Introduction

Dans la version du gz-buffer décrite ci-dessus, nous avons signalé le recours inévitable à quelques approximations et la présence de quelques limitations dues à un manque d'informations sur les entités de la scène déjà affichées. De plus, la méthode du gz-buffer impose la contrainte de ne traiter qu'un seul pixel de bord par ligne ou par colonne selon le cas.

Nous allons présenter ici une nouvelle méthode pour des scènes polygonales 3D [GHAZ 89] qui permet de s'affranchir de la plupart des obstacles signalés ci-dessus, et qui sont inévitables dans un contexte strictement 3D comme celui du gz-buffer. Cette méthode qui suit une démarche autre qu'un traitement des bords au fur et au mesure de l'affichage des polygones, est simple, générale et efficace . De plus, elle est rapide car les calculs les plus coûteux (antialiassage, mise en perspective de textures, ...) sont effectués uniquement pour les pixels visibles de la scène.

Comme le gz-buffer, cette méthode peut antialiasser des scènes 3D composées de polygones texturés, de couleurs uniformes ou en dégradés avec tout modèle d'illumination, et elle est bien adaptée à un environnement de tampon de profondeur cablé (ou microprogrammé). Cette méthode hybride peut utiliser une des méthodes incrémentales rapides pour le traitement des pixels de bord et une méthode de sur-échantillonnage local assurant, à un moindre coût, le traitement des pixels appartenant à des intersections de polygones. Cette méthode nécessite seulement un bit de mémoire supplémentaire par pixel associé aux bits de la mémoire d'images pour le codage des couleurs. Ce bit est réservé aux différents types de marquages ; ce peut être , par exemple, un bit dans les plans de superposition ("overlay") disponibles dans la plupart des systèmes graphiques ou le bit de poids fort dans les plans de la mémoire d'images. En tout cas, l'ajout ou la réservation d'un bit supplémentaire par pixel peuvent être considérés comme négligeables dans un système graphique qui dispose généralement de quelques dizaines de bits par pixel.

IV.4.4.2 Description de la méthode

Cette méthode, qui ne repose pas sur un antialiassage des bords au fur et à mesure de l'affichage des polygones, comporte trois étapes principales :

1- affichage de la scène et marquage des bords ;
2- détection des intersections visibles éventuelles et traitement d'antialiassage de ces intersections ;
3- traitement des pixels de bords visibles.

1- Marquage des bords :

La première étape de cette méthode consiste à marquer les bord visibles des polygones de la scène 3D en affichant celle-ci à l'aide d'un tampon de profondeur. Lors de l'affichage de chaque polygone p_j, on marque tout pixel visible de ses bords à l'aide du bit de marquage. Ces pixels marqués seront identifiés comme des pixels de bord pendant la phase

de traitement. Si le système comprend n bits par pixel pour coder les couleurs, le bit de marquage est considéré comme étant le (n+1)ième bit ; il sera utilisé, entre autres, pour le marquage des pixels de bord lors de cette étape. Avec cette définition, une valeur $0 \leq \mathbf{c} < 2^n$ correspond à une couleur pour un pixel interne de p_j ; la valeur $\mathbf{b}=2^n$ correspondra à un pixel de bord marqué.

Remarque :

Si le polygone p_j porte une texture, tous ses pixels internes peuvent être affichés avec une couleur quelconque **c**, $0 \leq \mathbf{c} < 2^n$. Signalons d'ailleurs, qu'il est possible d'afficher tous les pixels internes des polygones (avec ou sans texture) composants la scène avec la même valeur **c** lors de la première étape (cf. figure IV.6).

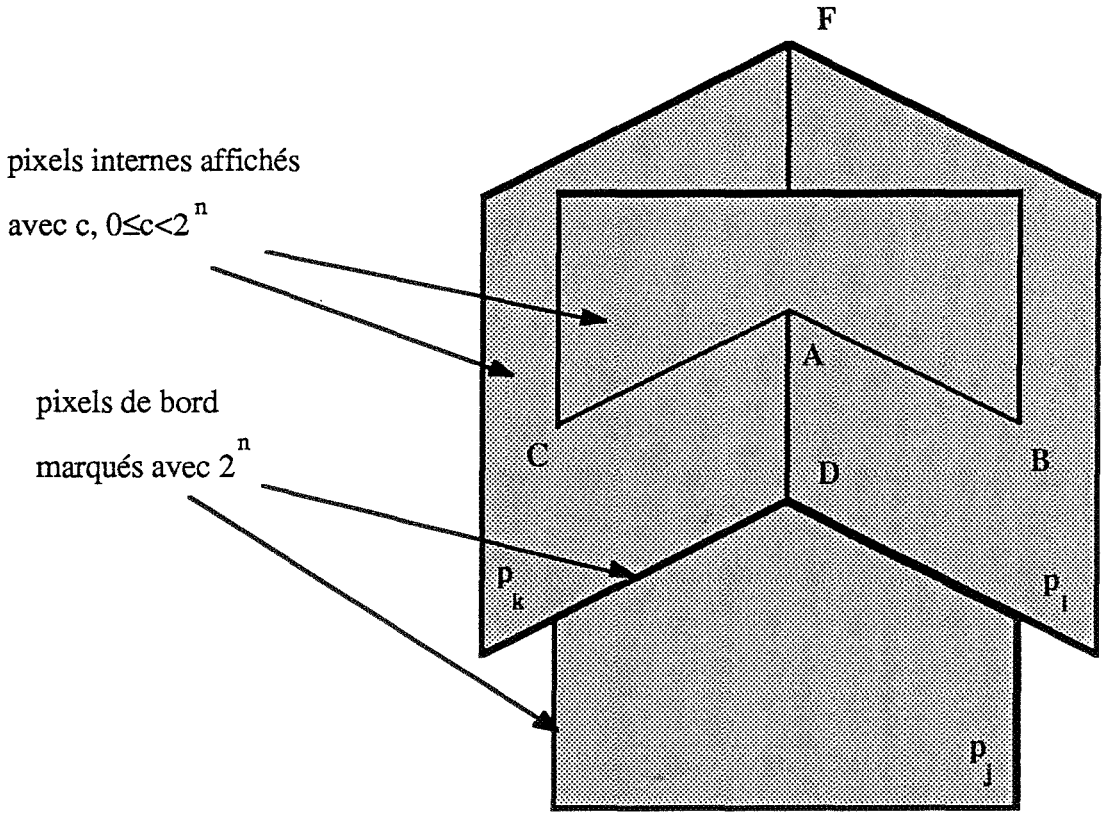


Figure IV.6 : Marquage des bords.

A la fin de cette première étape, tous les pixels visibles de la scène sont déterminés avant l'application de tout traitement d'antialiassage. Ceci nous permettra d'éviter les problèmes des polygones adjacents et du changement de couleur de fond dans un contexte plutôt 2D. De plus, grâce à cette étape, on n'applique les calculs les plus coûteux (comme la mise en perspective de textures ou le traitement des pixels de bord) que sur les pixels visibles de la

scène. Le gain de temps est considérable par rapport aux autres méthodes pour des scènes complexes comportant de nombreuses faces texturées cachées. Le problème majeur qui subsiste est la détection et le traitement des pixels situés sur des intersections entre polygones.

2- Détection et traitement des intersections :

Cette deuxième étape permettra de résoudre le problème de la détection et de l'antialiassage des pixels concernés par les intersections visibles entre polygones. Pour trouver les intersections, on peut utiliser un masque, en particulier si l'on dispose d'un affichage rapide , un tampon de profondeur cablé par exemple, afin d'éviter des calculs de profondeurs. Lors de cette étape, pour chaque polygone courant p_j, on marque ses pixels visibles et non marqués (les pixels d'une valeur comprise entre 0 et 2^n-1) avec la valeur **int**=2^n+1 en utilisant de nouveau le bit de marquage, à l'aide d'un tampon de profondeur. Ensuite, on va chercher les éventuelles intersections de p_j avec les autres polygones visibles de la scène (qui sont connus grâce à l'affichage de la scène lors de la la première étape) en balayant la boîte englobante de p_j.

Soit AB l'intersection entre les polygones p_j et p_l (cf. figure IV.7), où p_j est le polygone courant et p_l un polygone affiché postérieurement à p_j. Deux pixels, par ligne ou par colonne, peuvent être considérés comme les pixels concernés par l'intersection AB : celui du "côté" de p_j et celui du "côté" de p_l. Considérons un pixel (x,y) de valeur **int** appartenant à la fois à p_j et à AB. Ce pixel a quatre voisins directs : les pixels (x-1,y), (x+1,y), (x,y-1) et (x,y+1). Parmi ces voisins directs, il en existe au moins un, noté (x_v,y_v), qui a une valeur $0 \leq$ **c** $<2^n$ et appartenant à p_l (cf. figure IV.7). C'est une propriété spécifique des pixels d'intersection. Pour tout autre pixel de valeur **int** (pixel entièrement à l'intérieur de p_j), les quatre voisins directs ont une valeur supérieure ou égale à 2^n. Autrement dit, tout pixel entièrement à l'intérieur de p_j a des voisins directs qui sont soit d'autres pixels de valeur **int**, soit un pixel de bord de valeur **b**. Le pixel (x_v,y_v) défini ci-dessus est le deuxième pixel (avec (x,y)) concerné par l'intersection entre p_j et p_l. La boîte englobante de p_j est balayée afin de détecter les pixels sur toute intersection éventuelle appartenant à p_j (comme AB ou AC) en testant les pixels voisins directs.

Le traitement des deux pixels sur l'intersection AB nécessite la disponibilité de deux types d'informations : la couleur de chacun des deux polygones p_j et p_l en (x,y) et (x_v,y_v), et la répartition de ces deux pixels entre p_l et p_j.

L'information concernant la répartition des pixels (x,y) et (x_v,y_v) entre p_j et p_l peut être obtenue grâce à un sur-échantillonnage local. Comme nous le verrons plus tard, pour effectuer ce sur-échantillonnage, il faut comparer la profondeur de p_j et de p_l au centre de chaque sous-pixel des pixels (x,y) et (x_v,y_v) afin de déterminer le polygone qui est visible en ce point. Ceci nécessite la disponibilité du z de chacun des deux polygones p_j et p_l. Etant donné que p_l n'est pas le polygone courant, cette information est indisponible à cet instant précis de l'algorithme. Pour pouvoir effectuer le traitement d'antialiassage, nous marquons dans un premier temps le pixel (x_v,y_v) avec une valeur $2^n+1<\mathbf{i}_j<2^{n+1}$ en utilisant le bit de marquage (cf. algorithme IV.3). La valeur $\mathbf{i}_j-(2^n+2)$ est un indice dans un tableau, TAB, qui contient les coefficients de l'équation z=ax+by+c du plan des polygones du type p_j concernés par au moins une intersection visible. Par exemple, dans le cas de la figure IV.7, les coefficients a_j, b_j et c_j du plan du polygone p_j concerné par les intersections AB et AC seront stockés dans un tableau. Ceci nous permettra de connaître ces coefficients quand p_l (ou p_k) deviendra le polygone courant. Après avoir marqué tout pixel (x_v,y_v) sur les intersections AB et AC avec une valeur $\mathbf{i}_j$, nous aurons la situation suivante :

- tout pixel (x,y) de valeur **int** est un pixel interne de p_j, c'est à dire un pixel entièrement à l'intérieur de p_j ou un pixel d'intersection "côté" p_j;

- tout pixel (x,y) de valeur **b** est un pixel de bord ;

- tout pixel (x_v,y_v) de valeur $\mathbf{i}_j$ est un pixel d'une intersection, du type AB, entre p_j et un autre polygone, du type p_l.

Nous balayons à nouveau la boîte englobante de p_j ; à tout pixel interne de p_j (pixel de valeur **int**) on affecte la couleur de p_j en ce pixel : $coul(x,y)=p_j(x,y)$.

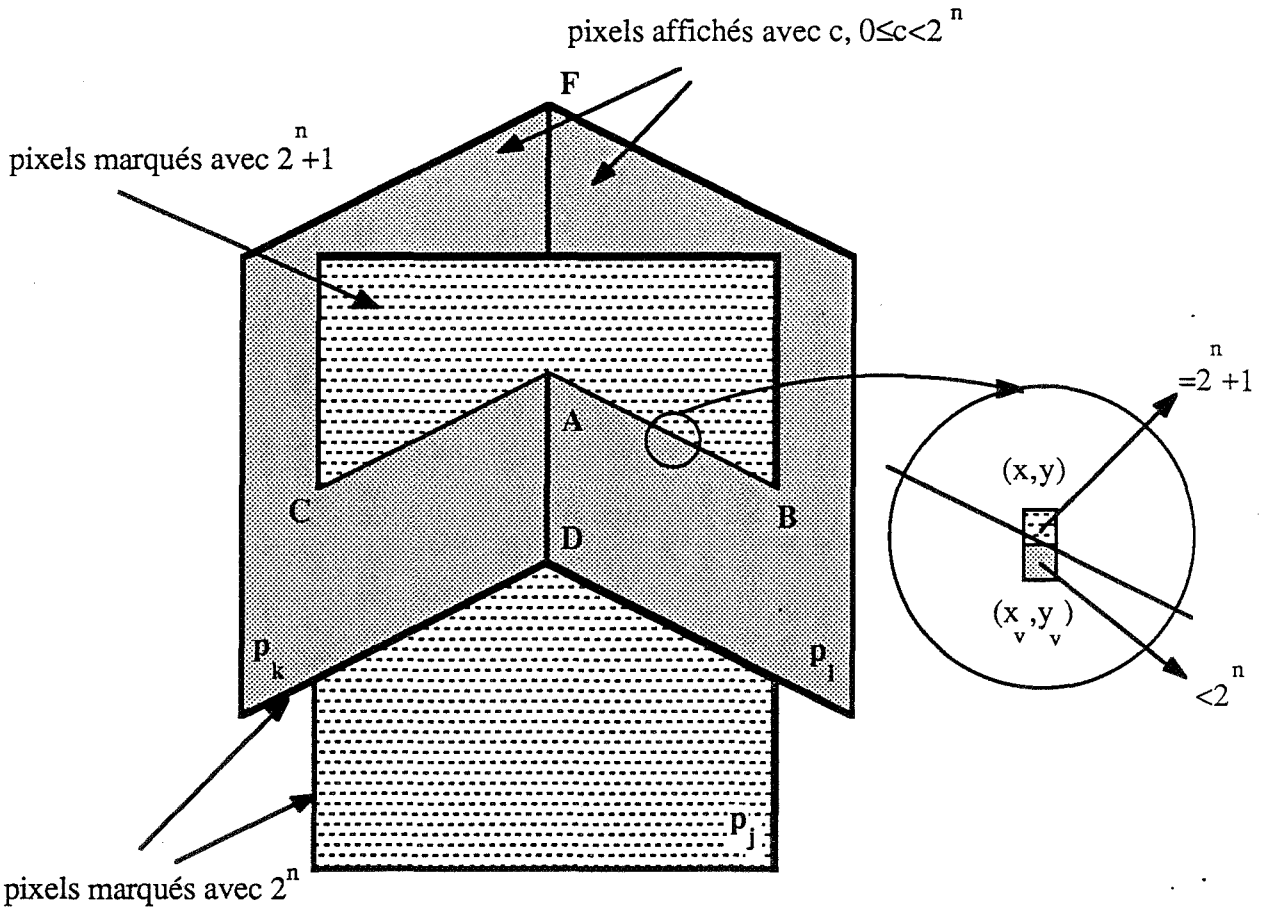


Figure IV.7 : Détection des pixels d'intersection.

Remarque :

Le marquage effectué sur les pixels d'intersection en vue d'un antialiassage ultérieur est indispensable. Contrairement au cas où la couleur d'un pixel peut être obtenue à l'aide de celle d'un pixel voisin, la profondeur de p_l au centre d'un pixel ou d'un pixel voisin ne peut pas être utilisée pour les sous-pixels appartenant à ce pixel, car cette profondeur ne permet pas de déterminer la visibilité des polygones p_j et p_l en ces sous-pixels. La figure IV.8 représente un cas où le polygone courant p_j est parallèle au plan xy et p_l est un polygone oblique par rapport à ce plan. Ce cas montre bien l'inefficacité de l'utilisation de la valeur de $z_l(x_c,y_c)$ pour les sous-pixels de ce pixel d'intersection car, en fonction des deux valeurs constantes z_j et $z_l(x_c,y_c)$, tous les sous-pixels seront faussement détectés comme appartenant soit à p_j, soit à p_l.

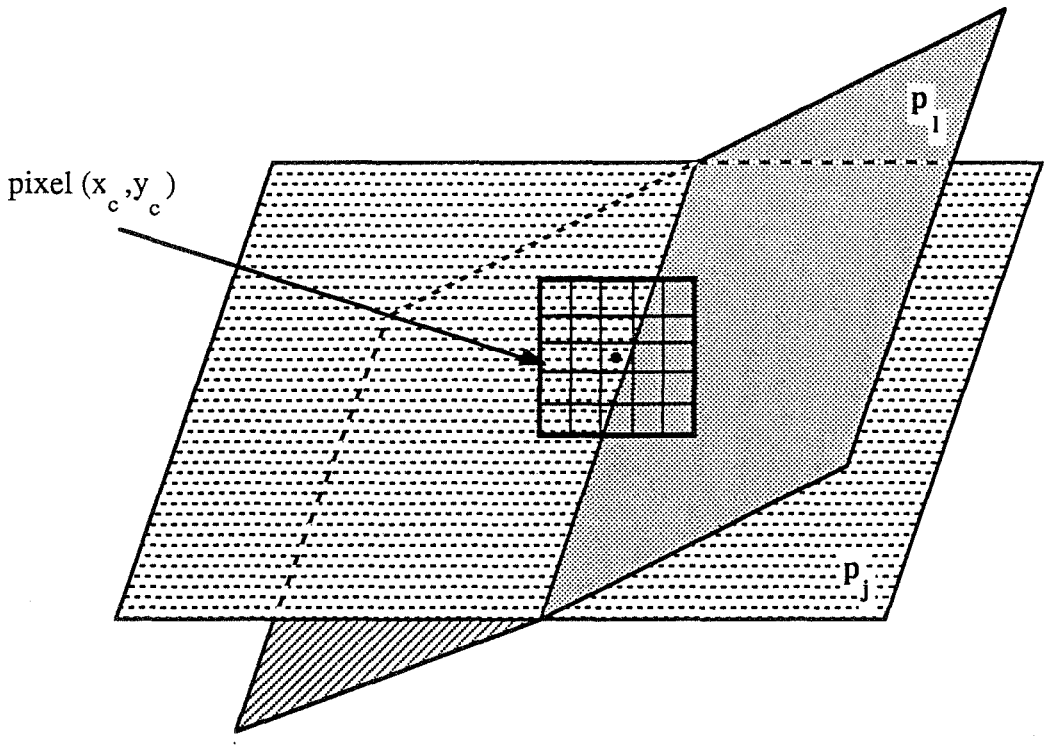


Figure IV.8 : Les problèmes de la répartition d'un pixel d'intersection.

Considérons maintenant le masquage de p_l quand il devient le polygone courant. Tous ses pixels visibles non-marqués (les pixels d'une valeur supérieure ou égale à 2^n sont déjà marqués) seront affichés avec la valeur **int**=2^n+1 (cf. figure IV.9). La boîte englobante de p_l est balayée. Les pixels sur l'intersection AB, déjà marqués quand p_j était le polygone courant, et leurs voisins concernés par AB seront détectés et traités. Si le pixel (x,y) est un pixel de l'intersection AB (coul(x,y)=$\mathbf{i}_j$), il sera détecté car la valeur de ce pixel est supérieure à 2^n+1 et qu'il a au moins un pixel voisin ayant la valeur **int** (cf. algorithme IV.3). L'autre pixel concerné par l'intersection AB est le voisin direct (x_v,y_v) qui a une valeur 0≤ **c**<2^n (cf. figure IV.9). Les coefficients a_l,b_l et c_l sont connus et a_j,b_j et c_j sont obtenus par le tableau des coefficients, TAB, avec l'indice $\mathbf{i}_j$-(2^n+2). Pour chaque sous-pixel (x_s,y_s) d'un pixel d'intersection, on peut donc calculer les valeurs de $z_j(x_s,y_s)$ et $z_l(x_s,y_s)$ et y déterminer la visibilité de chacun des deux polygones p_j et p_l. La couleur du polygone visible sera affectée au sous-pixel (x_s,y_s). Comme p_j n'est plus le polygone courant, on peut utiliser la couleur d'un pixel voisin approprié au lieu de la couleur de p_j en (x,y). Rappelons que, dans le cas d'un polygone de couleur uniforme, on trouve la couleur exacte et dans le cas d'un polygone texturé l'approximation est négligeable car la texture est déjà antialiassée par un filtrage passe-bas.

De cette manière, l'intersection AB peut être antialiassée par un sur-échantillonnage local. Si p_l a des intersections visibles avec d'autres polygones que p_j, ces nouvelles

intersections seront marquées de la même manière que l'intersection AB quand p_j était le polygone courant. A la fin, comme dans le cas du polygone p_j, à tout pixel interne de p_l (coul(x,y)=**int**) on affectera la couleur $p_l(x,y)$.

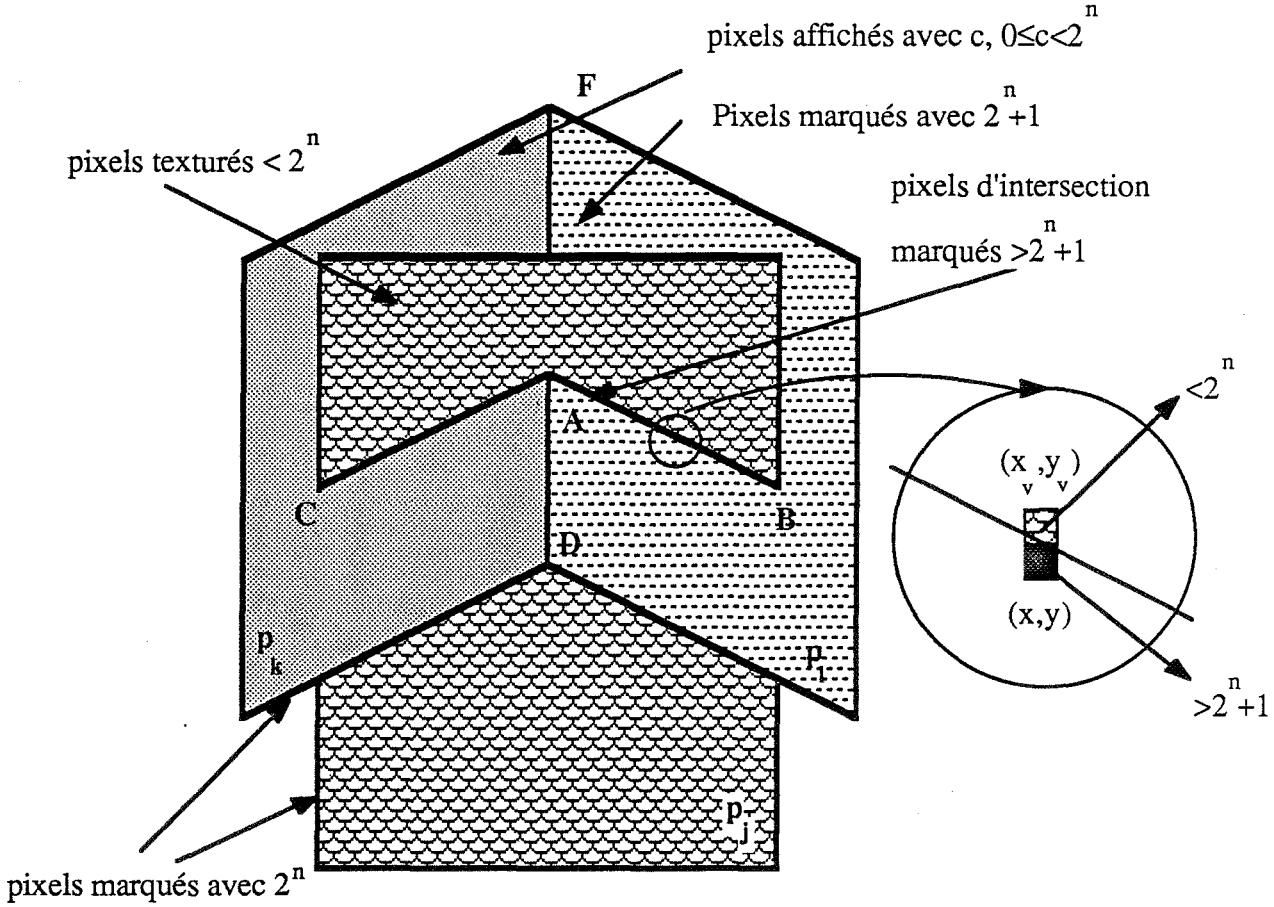


Figure IV.9 : Antialiassage des pixels d'intersection.

Le marquage et la détection des pixels d'intersections en vue d'un sur-échantillonnage local ne nécessitent qu'un buffer de taille trois lignes d'image, pour effectuer les tests sur les pixels voisins d'un pixel donné, et un tableau de coefficients, TAB, de taille 3p, où p vérifie la relation $0 \leq p < s$ si s est le nombre de polygones ayant au moins une intersection visible avec un autre polygone. La valeur de p peut dépendre de l'ordre d'affichage des polygones ayant au moins une intersection visible. En général, p est assez proche de $\frac{s}{2}$. Il peut cependant, dans le pire des cas, être égal à s-1.

3- Traitement des pixels de bord :

Après avoir antialiassé toutes les intersections, nous allons traiter les bords visibles des polygones qui contiennent les pixels marqués par la valeur **b** (cf. figure IV.10). Une

méthode d'antialiassage incrémentale rapide ayant une moyenne pondérée pour le filtrage passe-bas comme celle de [GUPT 81] peut être employée en utilisant la couleur d'un pixel voisin approprié pour le traitement de chaque pixel de bord et en respectant un sens de parcours des bords de polygone.

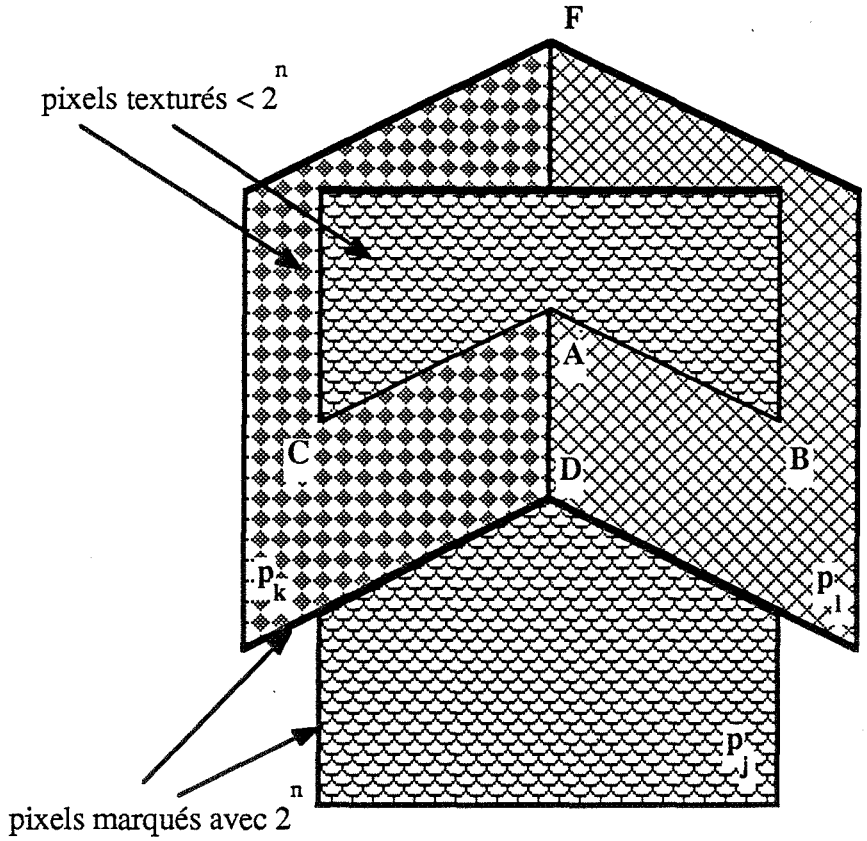


Figure IV.10 : Antialiassage des bords marqués.

Signalons que le problème du changement de couleur de fond ne se pose pas avec notre méthode : ce problème est en effet lié à un traitement des bords au fur et au mesure de l'affichage des polygones. Notre méthode, grâce à un contexte plutôt 2D, évite ce problème qui est le plus difficile à résoudre avec les techniques proposées auparavant. De plus, le problème des polygones adjacents a été évité car lors du traitement des bords, la couleur des deux côtés de chaque pixel est connue et les bords en commun ne sont traités qu'une seule fois grâce à leur marquage.

Les principes de cette méthode sont résumés par l'algorithme suivant.

Algorithme IV.3, la méthode en étapes successives :

```
{
/* Marquage des pixels de bords : */
pour chaque polygone p_j de la scène faire
 si pixel (x,y) est visible alors
  si pixel (x,y) est interne alors coul(x,y):=c;   /* 0≤c<2^n */
  sinon coul(x,y):=b=2^n;

/* Détection, marquage et antialiassage des intersections : */
indice_écriture:=0;
pour chaque polygone p_j de la scène faire
{
intersection:=faux;
calculer a_j, b_j et c_j;  /* coefficients de l'équation du plan p_j */
afficher tous les pixels visibles non-marqués de p_j avec la valeur int:=2^n+1;
pour chaque pixel (x,y) de la boîte englobante de p_j faire
{
 si coul(x,y)=int alors
  {
   si pour un des pixels voisins directs, (x_v,y_v), coul(x_v,y_v)<2^n alors
    { /* Marquage de l'intersection */
     coul(x_v,y_v):=i_j=indice_écriture+2^n+2;   /* 2^n+1<i_j<2^n+1 */
     intersection:=vrai;
    }
  }
 sinon
  si coul(x,y)>2^n+1 et il existe un pixel voisin de valeur int alors
    {  /* (x,y) est un pixel (déjà marqué) d'intersection entre p_j et un polygone p_i affiché auparavant */
     trouver le voisin direct (x_v,y_v) vérifiant 0≤coul(x_v,y_v)<2^n;
     indice_lecture:=coul(x,y)-(2^n+2);
     trouver a_i,b_i et c_i dans le tableau des coefficients TAB;
     subdiviser les pixels (x,y) et (x_v,y_v) en sous-pixels;
     pour chaque sous-pixel (x_s,y_s) faire
      {
       z_j(x_s,y_s):=a_j x_s+b_j y_s+c_j;
       z_i(x_s,y_s):=a_i x_s+b_i y_s+c_i;
       si z_j(x_s,y_s)<z_i(x_s,y_s) alors coul(x_s,y_s):=p_j(x,y);
       sinon coul(x_s,y_s):=p_i(x_v,y_v);
      }
     calculer la couleur finale des pixels (x,y) et (x_v,y_v) par un filtrage passe-bas;
    }
}
si intersection=vrai alors
 {
  mettre a_j, b_j et c_j dans le tableau de coefficients TAB;
  indice_écriture:=indice_écriture+1;
 }
pour chaque pixel (x,y) de la boîte englobante de p_j faire
si coul(x,y)=int alors coul(x,y):=p_j(x,y);
}

/* Antialiassage des pixels de bords : */
pour chaque polygone de la scène faire
antialiasser les bords visibles marqués b à l'aide de l'algorithme incrémental de
```

Gupta et Sproull en respectant un sens de parcours des polygones et en utilisant la couleur des pixels voisins appropriés;
}

IV.4.4.3 Conclusion

La méthode proposée ci-dessus résout les trois types de problèmes exposés dans le paragraphe IV.3. Plus particulièrement, elle peut résoudre facilement le problème de la détection et de l'antialiassage correct des pixels d'intersection qui ne sont pas toujours situés sur un segment de droite discrétisé, contrairement à ce que l'on pouvait attendre, à cause de l'imprécision du tampon de profondeur. L'image IV.4 représente les résultats obtenus avec cette méthode pour une scène non-texturée ayant les trois types de problèmes d'antialiassage avec le tampon de profondeur ; l'image IV.1 est l'affichage de la même scène sans antialiassage et l'image IV.2 est sa visualisation avec un antialiassage 2D. L'image IV.6 (correspondant à la figure IV.6) montre les résultats obtenus avec cette méthode pour une scène texturée. D'autre résultats de cette méthode sont représentés par l'image IV.8 ; l'image IV.7 montre l'affichage de la même scène sans antialiassage.

Un seul bit supplémentaire, ou réservé, par pixel dans un système graphique et peu de place mémoire utilisée (trois lignes d'image et un tableau de petite taille), rendent cette méthode particulièrement intéressante, même en la comparant au gz-buffer. Les opérations de bases nécessaires sont essentiellement les opérations classiques existantes dans les systèmes graphiques :

- calcul et comparaison de profondeurs, des opération identiques à celles d'un tampon de profondeur conventionnel ;

- marquage des pixels, opération d'écriture ;

- test des pixels, opération de lecture.

Le sur-coût matériel raisonnable (cf. figure IV.11) et la simplicité de cette méthode peut donc permettre facilement une implantation matérielle.

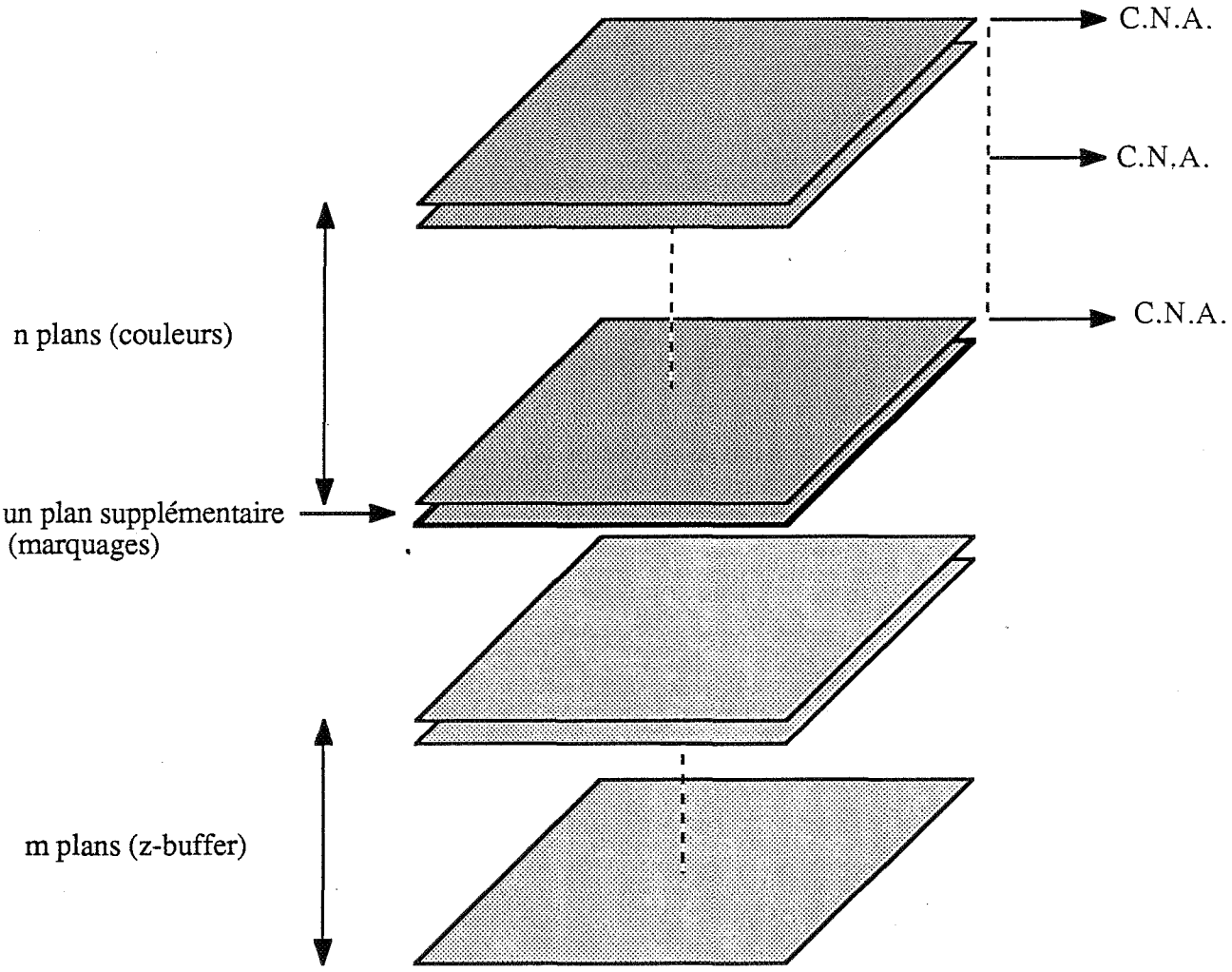


Figure IV.11 : Configuration des mémoires.

Grâce à la première étape, cette méthode applique les calculs les plus coûteux comme la mise en perspective des pixels ou l'antialiassage uniquement sur les pixels visibles de la scène. C'est un avantage non négligeable par rapport aux autres méthodes notamment pour des scène complexes. Dans la plupart des cas, cette méthode est plus rapide que le gz-buffer (qui est une méthode assez rapide). Dans le cas d'une scène 3D simple et sans texture présentée par l'image IV.4, cette méthode est 2 fois plus rapide que la version du gz-buffer décrite ci-dessus et 4 fois plus lente que l'application directe de la méthode 2D de Gupta et Sproull.

Pour les bords visibles, cette méthode utilise un antialiassage rapide par une méthode incrémentale. Par conséquent, le problème de petits objets ne peut pas être résolu, les méthodes incrémentales ne pouvant antialiasser les petits objets (cf. chapitre III).

Signalons enfin qu'on peut éviter le masquage en affichant chaque polygone de la scène avec sa propre couleur (ou texture) lors de la première étape. Ceci nécessite essentiellement la comparaison classique des profondeurs. Cependant, il est intéressant d'utiliser un masque notamment si l'on dispose d'un affichage rapide par un tampon de profondeur cablé.

IV.4.5 Conclusion

Nous pouvons diviser les méthodes d'antialiassage pour le tampon de profondeur en trois catégories :

- Les méthodes telles que [CARP 84] et [SALE 89] sont assez précises et peuvent produire des images de bonne qualité graphique, mais elles n'ont pas la simplicité du tampon de profondeur qui est à l'origine de sa popularité. Ces méthodes ont besoins de structures de données complexes et d'un espace mémoire très important. Malgré l'utilisation d'un tampon de profondeur, quelques tris préalables ou des affichages par balayage de ligne sont souvent nécessaires. Enfin, ces méthodes sont assez lentes.

- Les méthodes telles que [EVAN 84] et [DUFF 85] sont assez simples mais elles restent approximatives et elles ne peuvent être utilisées que dans des cas restreints (pas de changement de couleur de fond, par exemple). En effet, dans le cas général, elles ne peuvent pas résoudre les trois types de problèmes d'aliassage survenant avec le tampon de profondeur.

- Les méthodes développées dans [GHAZ 85], [GHAZ 87] et [GHAZ 89] sont en général simples et rapides ; elles peuvent résoudre, dans la plupart des cas, les trois types de problèmes d'aliassage liés à l'utilisation du tampon de profondeur. Cependant, quelques petites approximations et quelques cas particuliers subsistent dans ces méthodes.

IV.5 Antialiassage des ombres portées générées par le tampon de profondeur

L'utilisation des ombres portées dans les images de synthèse est un élément qui améliore sensiblement leur réalisme, notamment pour les informations de profondeur ainsi introduites. On peut trouver diverses techniques d'élimination des parties cachées qui

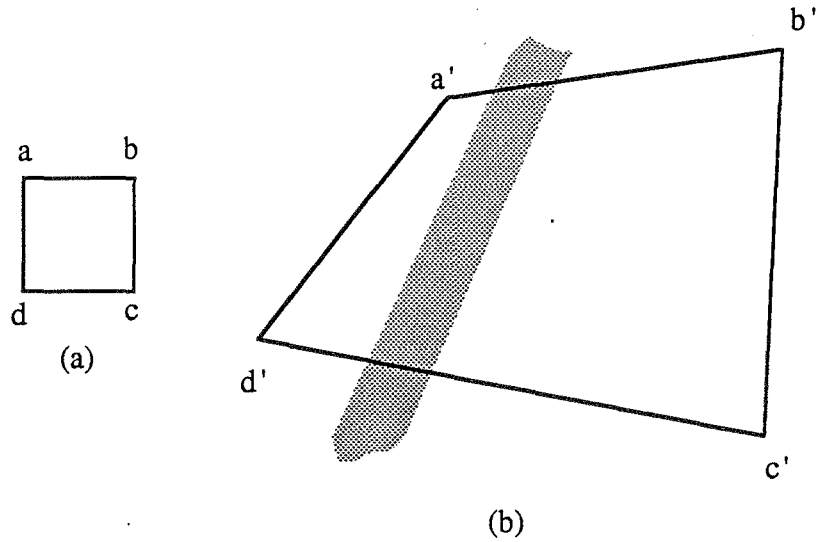
permettent l'affichage des ombres portées. Une classification de celles-ci en cinq techniques de base est donnée dans [MAX 86]. Les ombres portées peuvent facilement être générées par la technique du tampon de profondeur [WILL 78] sans aucune restriction sur les objets composant la scène. La méthode de Williams utilise un double tampon de profondeur : un tampon de profondeur conventionnel pour l'élimination des parties cachées et un autre pour une image vue de la source lumineuse ponctuelle assimilée à l'œil de l'observateur. L'algorithme comporte deux étapes : une première étape pour générer une image vue de la source lumineuse, image auxiliaire, qui détermine les pixels en ombre et une deuxième étape pour éliminer les parties cachées vues de l'observateur. L'éclairage de chaque pixel de l'image finale est défini par une transformation appropriée et à l'aide de l'image auxiliaire. Cette méthode peut être utilisée pour plusieurs sources lumineuses, moyennant un tampon de profondeur par source lumineuse. La modification de la méthode initiale permet de tenir compte des réflexions ou des sources lumineuses non-ponctuelles [BROT 84].

Cette méthode très simple, générale et efficace n'est pourtant pas sans inconvénients, en particulier de très sérieux problèmes d'aliassage. Un pixel de l'image finale peut correspondre à une grande zone de l'image auxiliaire. En plus des problèmes habituels d'aliassage dus à la discrétisation au niveau de l'image auxiliaire, il y a une accentuation importante de ceux-ci à cause de la transformation des pixels d'une image à l'autre. Le premier problème est moins important que le second car l'aliassage induit par la partition des pixels de l'image auxiliaire situés sur la frontière de l'ombre est moins considérable que celui dû à l'échantillonnage ponctuel d'une zone composée de plusieurs pixels de l'image auxiliaire. Ce dernier problème est similaire à celui étudié dans le cas de la mise en perspective des textures (cf. chapitre V), mais les techniques d'antialiassage utilisées sont différentes. Le premier problème, sans grande importance pour le résultat final, est souvent non-traité. Nous étudierons quelques solutions pour le deuxième problème dû à la transformation des pixels entre l'image finale et l'image auxiliaire.

Le sur-échantillonnage global est la première solution d'antialiassage envisageable pour résoudre ce problème. Comme dans le cas de la mise en perspectives des textures, outre le temps de calcul très élevé, l'efficacité de cette méthode peut être limitée dans certains cas (cf. chapitre V). Nous proposons une solution de sur-échantillonnage local et adaptatif qui est similaire à celle décrite pour le tracé de rayons (cf. chapitre III) avec le risque de quelques approximations dans certains cas.

Nous remplaçons les centres des pixels de la grille d'affichage par leurs coins pour les calculs d'ombres portées. Etant donné que chaque coin est commun à quatre pixels voisins, le sur-coût de calcul est équivalent à celui d'une ligne et d'une colonne supplémentaires.

Si les quatre coins d'un pixel de l'image finale sont tous éclairés (ou tous situés dans l'ombre), on peut le considérer comme un pixel entièrement éclairé (ou situé dans l'ombre). Mais cette méthode est approximative car on peut trouver un résultat erroné si la transformation d'un pixel considéré comme entièrement éclairé (ou dans l'ombre) correspond à une zone grande sur l'image auxiliaire et si cette zone contient au moins une bande d'ombre (ou éclairée) qui n'intersecte aucun coin de cette zone (cf. figure IV.12).

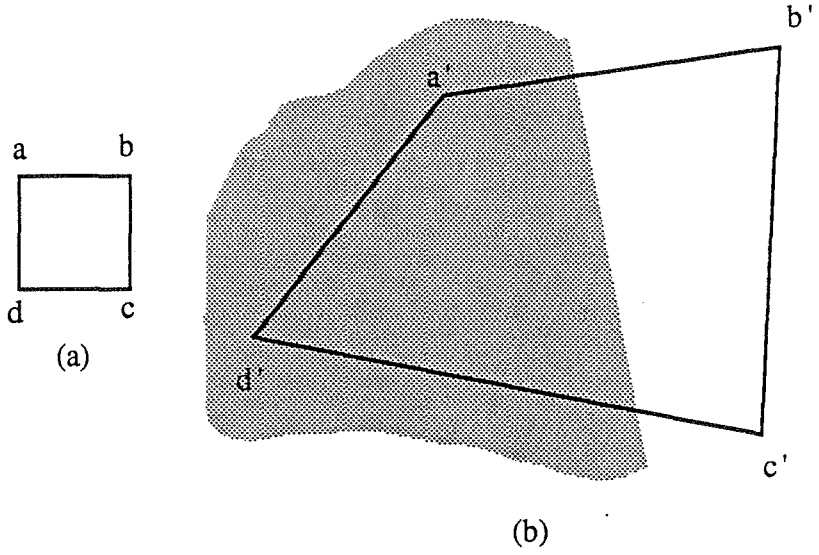


a) un pixel de l'image finale

b) la transformation d'un pixel contenant une bande d'ombre sur l'image auxiliaire

Figure IV.12 : La transformation d'un pixel à problème de l'image finale.

Si les quatre coins d'un pixel ne sont pas éclairés (ou situés dans l'ombre), ce pixel est partiellement éclairé (cf. figure IV.13) et un calcul plus détaillé s'impose. Dans ce cas, on subdivise récursivement un pixel en quatre carrés (cf. figure IV.14) et pour chaque carré on détermine si les quatre coins sont éclairés (ou dans l'ombre) ou pas. Dans le cas d'un carré partiellement éclairé, on continue l'opération jusqu'à ce que on trouve soit un carré entier soit un carré de taille inférieure à la limite fixée.



a) un pixel de l'image finale
b) la transformation d'un pixel partiellement éclairé sur l'image auxiliaire

Figure IV.13 : La transformation d'un pixel partiellement éclairé de l'image finale.

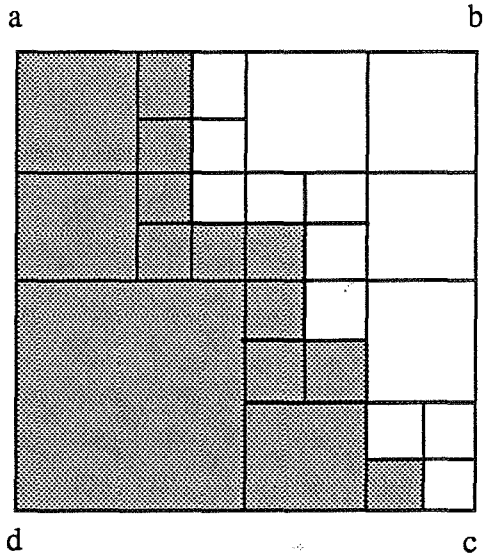


Figure IV.14 : La sous-division récursive d'un pixel partiellement éclairé de l'image finale.

La technique d'échantillonnage stochastique (cf. chapitres I et II) peut être considérée comme une méthode assez adaptée à l'antialiassage dans le cas des ombres portées. Une méthode utilisant un échantillonnage stochastique avec des bons résultats graphiques est présentée dans [REEV 87].

Références du chapitre IV

[BEIG 86] : M. BEIGBEDER, D. GHAZANFARPOUR and B. PEROCHE,
"The gz-buffer Method for Antialiasing", Deuxième colloque international de l'image électronique, CESTA, Nice, avril 1986, pp. 817,822.

[BROT 84] : L. BROTMAN and N. BADLER,
"Generating Soft Shadows with a Depth Buffer Algorithm", IEEE Computer Graphics and Applications, October 1984, pp. 5-12.

[CARP 84] : L. CARPENTER,
"The A -buffer, an Antialiased Hidden Surface Method", Computer Graphics, vol. 18, No. 3, July 1984, pp. 103-108.

[CATM 74] : E. CATMULL,
"A Subdivision Algorithm for Computer Display Curved Surfaces", Ph.D. thesis, University of Utah, Dec. 1974, pp. 40-49.

[CATM 78] : E. CATMULL,
"A Hidden-Surface Algorithm with Anti-Aliasing", Computer Graphics, vol. 12,No. 3, august 1978, pp. 6-11.

[DUFF 85] : T. DUFF,
"Compositing 3-D Rendered Images", Computer Graphics, vol. 19, No. 3, July 1985, pp. 41-43.

[EVAN 84] : K. EVANS,
"An Approximate Method for Antialiasing, Using a Random Access z-buffer", Graphics Interface, 1984, pp. 109.

[FUJI 84] : A. FUJIMOTO, G. PERROTT and K.IWATA,
"A 3-D Graphics Display system with Depth Buffer and pipeline Processor", IEEE Computer Graphics and Applications, June 1984, pp. 11-23.

[GHAZ 85] : D. GHAZANFARPOUR,
"Synthèse d'images et antialiassage", Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure des Mines, Saint-Etienne, novembre 1985.

[GHAZ 87] : D. GHAZANFARPOUR and B. PEROCHE,
"A Fast Antialiasing Method with z-buffer", Eurographics, 87, Amsterdam, August 1987, pp. 503-512.

[GHAZ 89] : D. GHAZANFARPOUR and B. PEROCHE,
"Antialiasing by Successive Steps with a z-buffer", Eurographics' 89, Hambourg, September 1989.

[GUPT 81] : S. GUPTA and R. SPROULL,
"Filtering Edges for Gray-Scale Displays", Computer Graphics, vol. 15, No. 3, august 1981, pp. 1-5.

[LANE 80] : J. LANE, L. CARPENTER, T. WHITTED and J. BLINN,
"Scan-line methods for displaying parametrically defined surfaces", Comm. ACM, vol. 25, No. 1, 1980, pp. 23-34.

[MAX 86] : N. MAX,
"Atmospheric Illumination and Shadows", Computer Graphics, vol. 20, No. 4, august 1986, pp. 269-278.

[PITT 80] : M. PITTEWAY and D. WATKINSON,
"Bresenham's Algorithm with Grey Scale", Communications of the ACM, vol. 23, No. 11, Nov. 1980, pp. 625-626.

[PORT 84] : T.PORTER and T. DUFF,
"Compositing Digital Images", Computer Graphics, vol. 18, No. 3, July 1984, pp. 253-259.

[REEV 87] : W. REEVES, D. SALESIN and R. COOK,
"Rendering Antialiased Shadows with Depth Maps", Computer Graphics, vol. 21, No. 4, July 1987, pp. 103-108.

[SALE 89] : D.SALESIN and J. STOLFI,
"The zz-buffer : A Simple and Efficient Rendering Algorithm With Reliable Antialiasing", PIXIM 89, Paris, September 1989, pp. 451-465.

78] : L. WILLIAMS,
g Curved Shadows on Curved Surfaces", Computer Graphics, vol. 12, No. 3, July
p. 270-274.

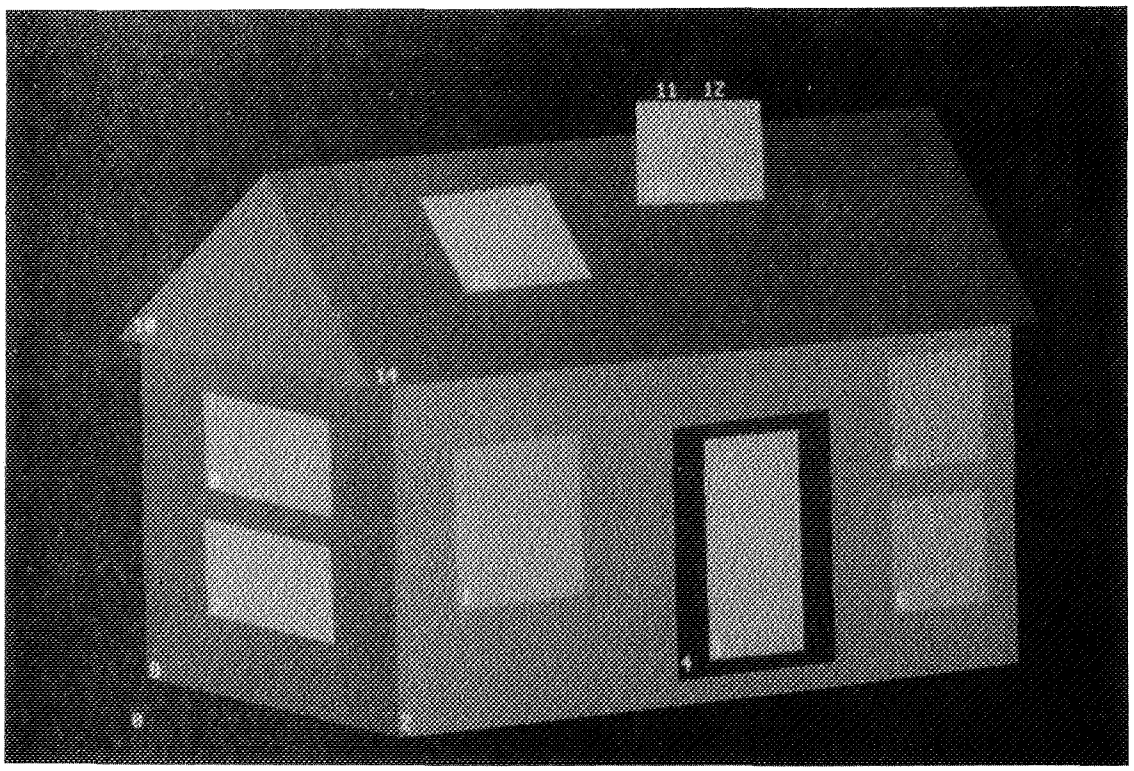


Image IV.1 : Image non-traitée d'une scène polygonale simple représentant les trois types de problèmes d'antialiassage avec le tampon de profondeur. Les numéros indiquent l'ordre d'affichage.

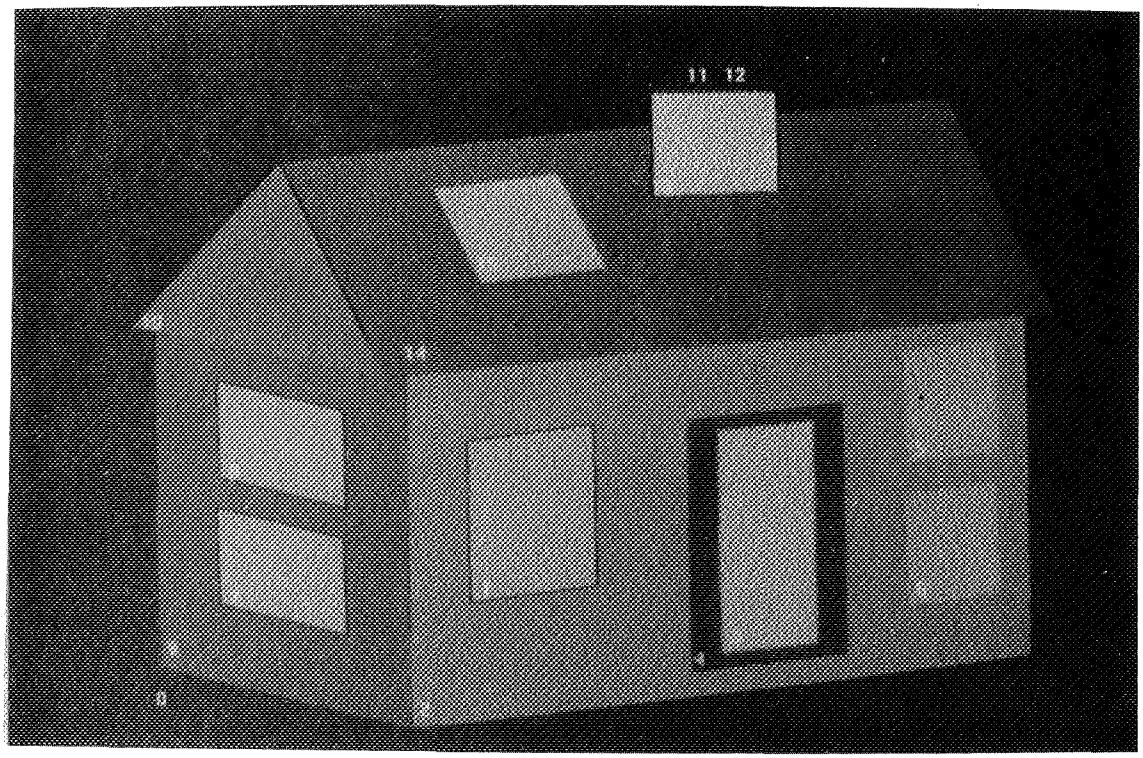


Image IV.2 : Image traitée par une méthode d'antialiassage 2D au fur et à mesure de l'affichage des polygones.

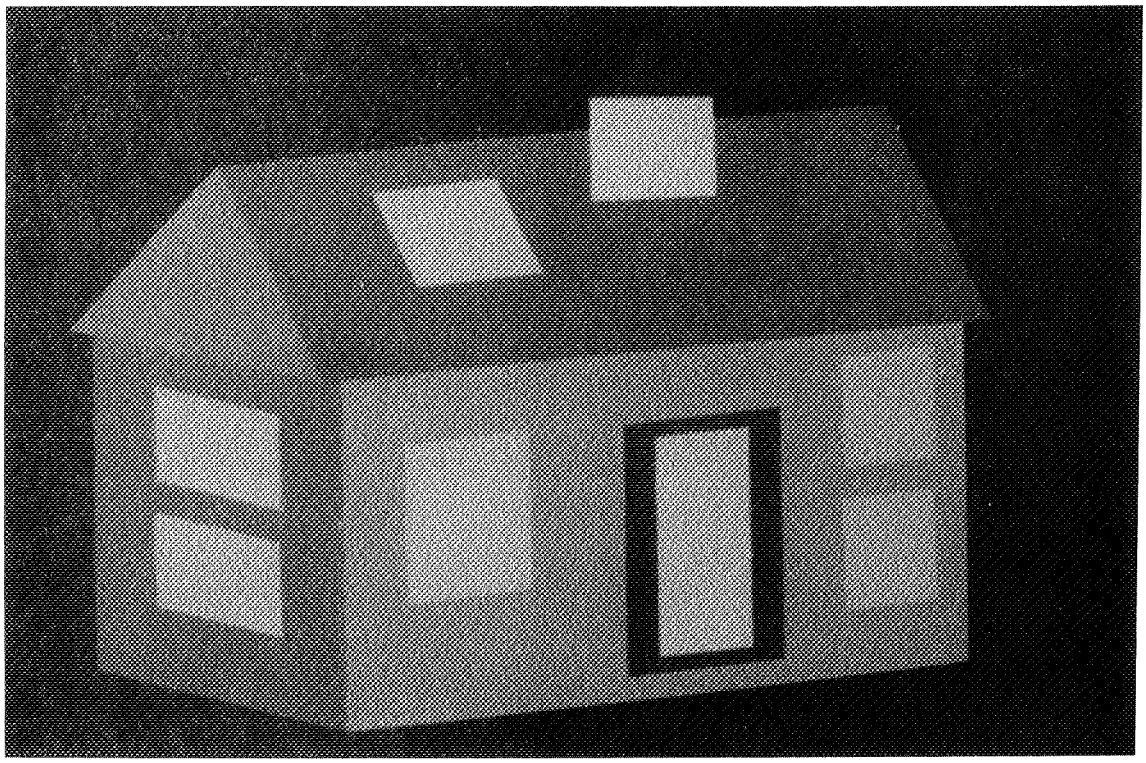


Image IV.3 : Image traitée par la méthode du gz-buffer.

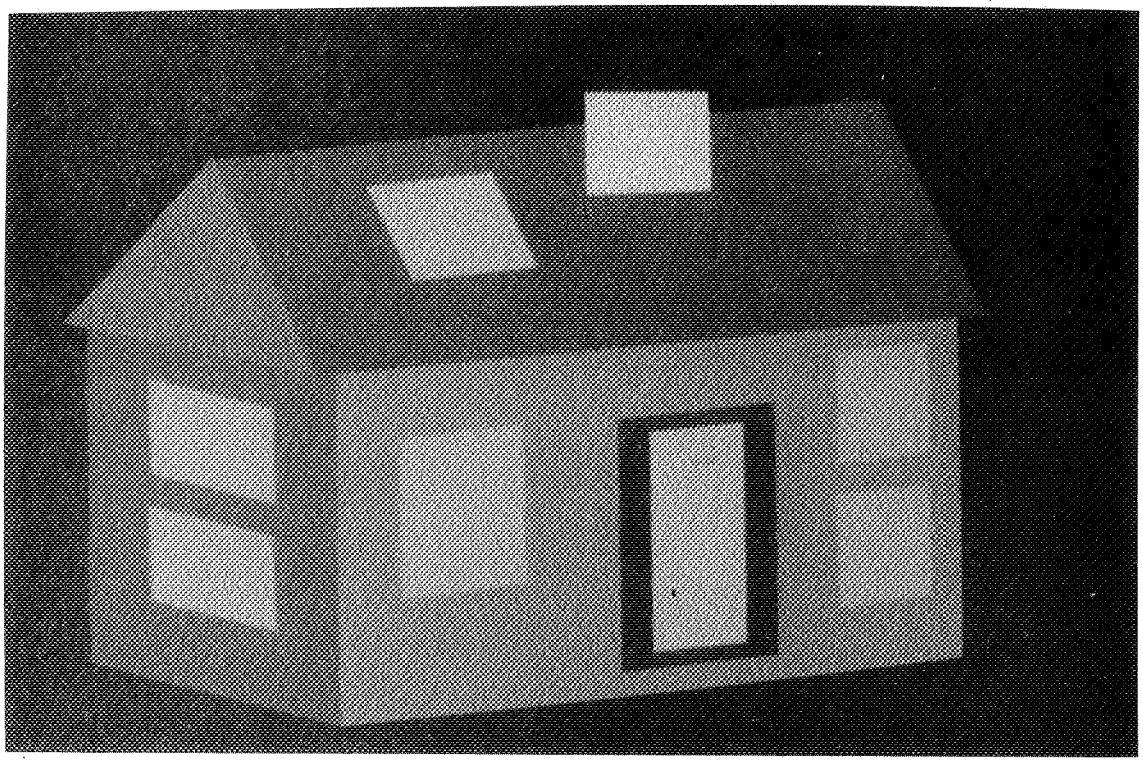


Image IV.4 : Image traitée par la méthode hybride en étapes successives.

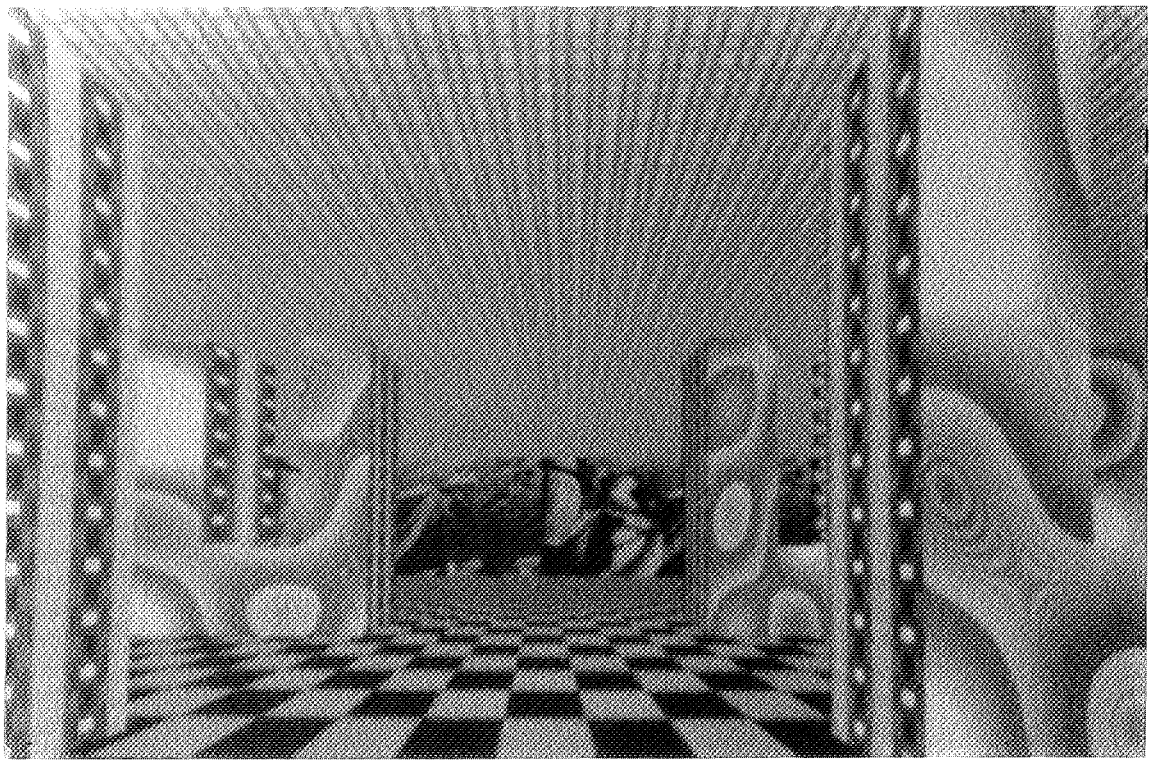


Image IV.5 : Image traitée par la méthode du gz-buffer.

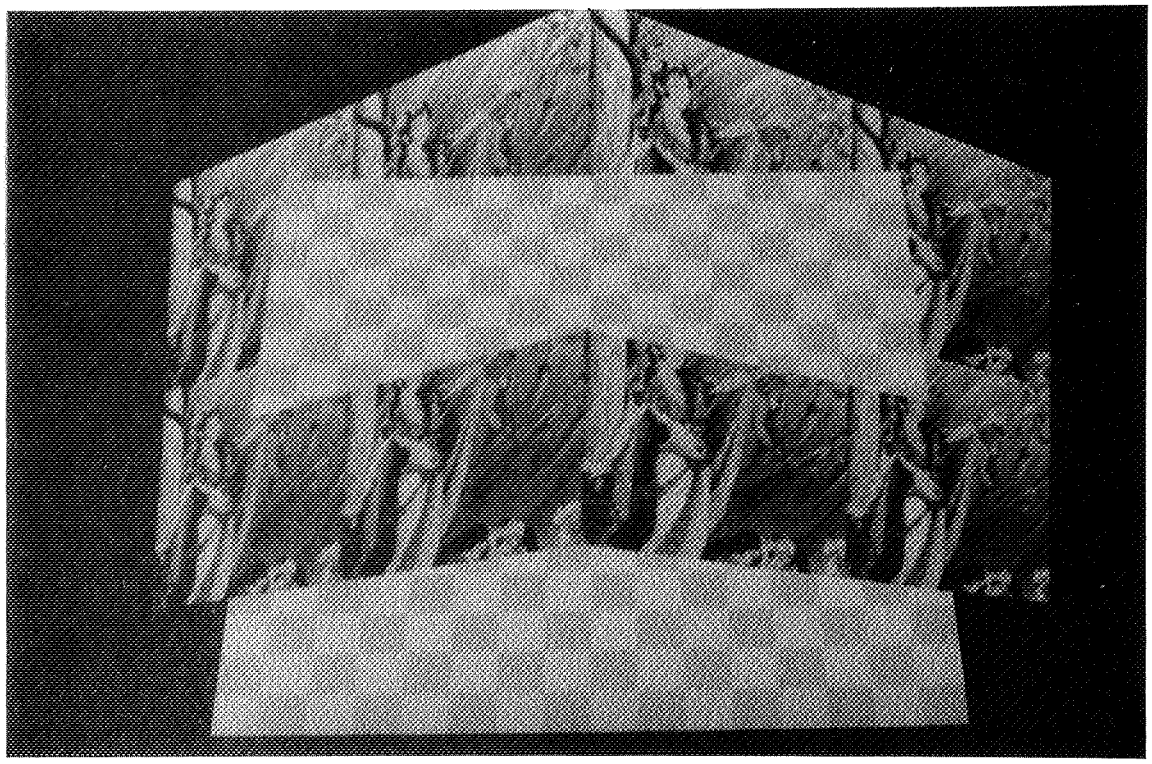


Image IV.6 : Image traitée par la méthode hybride en étapes successives.

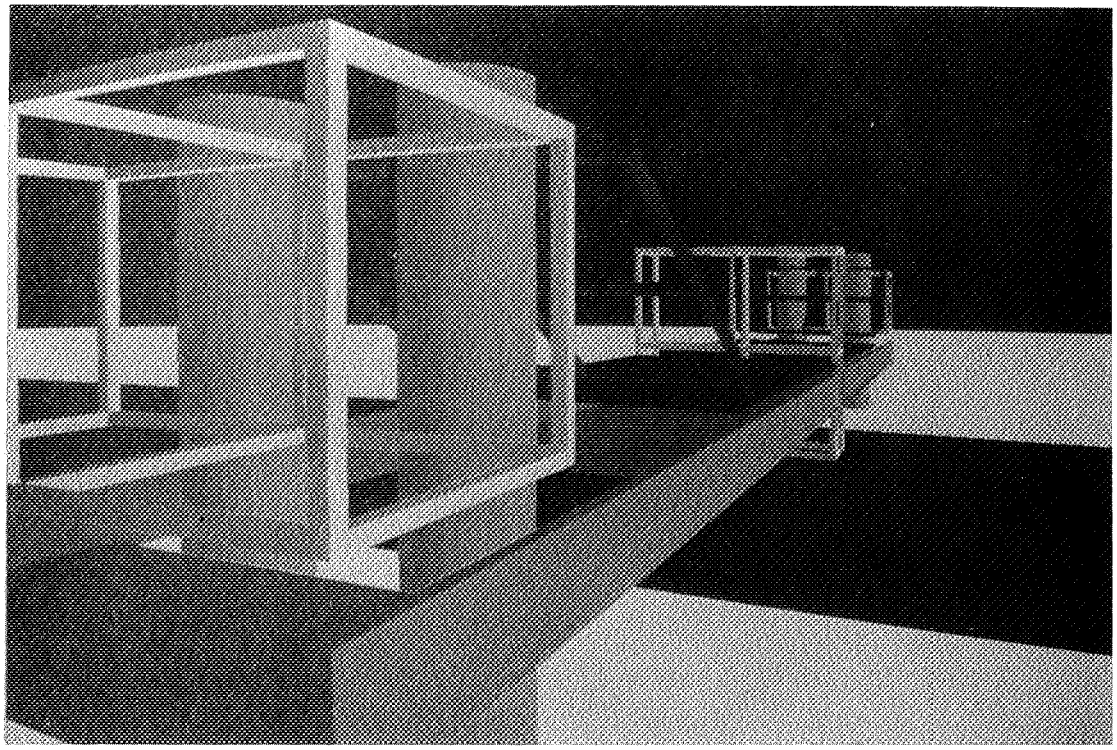


Image IV.7 : Image non-traitée.

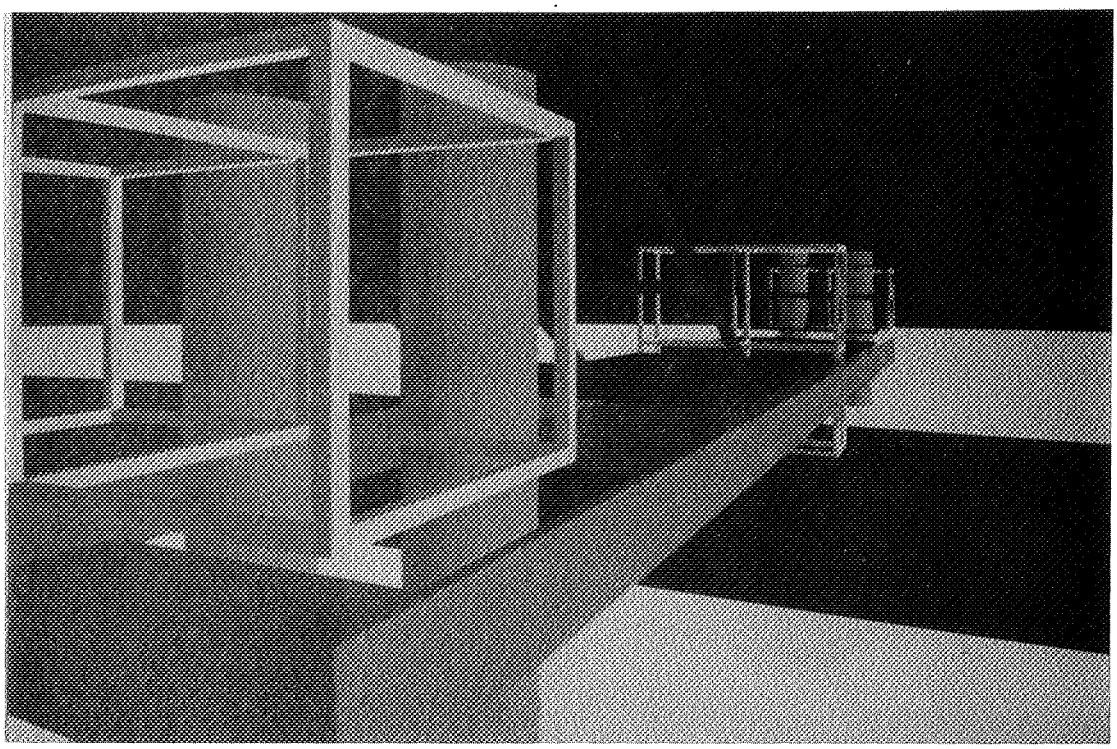


Image IV.8 : Image traitée par la méthode hybride en étapes successives.

CHAPITRE V

Textures et moirés

CHAPITRE V

Textures et moirés

V.1 Introduction

Le problème de la représentation des matériaux dans les images de synthèse apparaît dès que l'on souhaite visualiser des scènes 3D modélisées par des surfaces auxquelles est affectée non pas une couleur uniforme ou des dégradés de couleur, mais une couleur spatialement variable appelée texture. La mise en perspective des textures peut être utilisée avantageusement dans de nombreux domaines tels que la C.A.O. ou la simulation. L'analyse et la synthèse d'images ont deux définitions différentes des textures. Pour l'analyse d'images, une texture est l'aspect d'une surface à l'intérieur d'un contour ; une texture vérifie donc certaines conditions comme l'homogénéité, ou l'invariance statistique par translation. Pour la synthèse d'images, une texture est généralement considérée comme une image quelconque (cf. chapitre II), qui est collée sur une surface plane ou gauche, tel un papier peint sur un mur. Les termes d'image et de texture sont donc souvent confondus en synthèse d'images.

La présence de textures dans les images de synthèse est sans doute un des éléments les plus significatifs permettant d'obtenir des images d'un grand réalisme. En effet, en plus des détails de couleurs, les textures peuvent apporter à une surface les détails de formes qui ne sont pas toujours faciles à obtenir par une modélisation géométrique directe. Les premières utilisations des textures dans des images de synthèse remontent aux travaux de Catmull [CATM 74] en 1974 dans lesquels des textures naturelles (images) numérisées ont été plaquées sur des surfaces. Ces travaux furent généralisés par ceux de Blinn et Newell [BLIN 76] en 1976 utilisant des textures synthétiques périodiques et un modèle d'éclairement spéculaire. En synthèse d'images, deux types de textures sont utilisés :

- Les textures planes appelées textures 2D qui sont définies à plat et sont ensuite plaquées sur les surfaces des objets. Dans ce cas, une texture peut être considérée comme une image numérique, d'origine naturelle ou synthétique, plaquée sur une surface plane ou gauche. De plus, on peut prendre en compte facilement les différents attributs de la surface comme l'éclairement et la transparence. Les textures planes sont largement utilisées en synthèse d'images.

- Les textures volumiques ("solid textures") dites textures 3D sont définies pour tous les points de l'espace 3D. Les objets sont ensuite sculptés dans l'espace de la texture. A cause de sa complexité, cette technique a une utilisation beaucoup plus restreinte que la méthode précédente à l'heure actuelle.

Les deux types de textures peuvent provoquer le phénomène d'aliassage, essentiellement sous forme de moirés qui est un défaut très visible (cf. images V.3.a et V.5.a). Etant données l'importance et la multiplicité des textures planes, nous allons étudier en détail leur utilisation en synthèse d'images.

Dans ce chapitre, nous allons étudier les transformations géométriques (placage et projection perspective) sur les textures planes et l'effet de ces transformations sur l'image finale dans le domaine spatial et sur son spectre dans le domaine fréquentiel afin d'expliquer le phénomène d'aliassage qui est le problème majeur à résoudre dans ce cas. Après avoir présenté le problème, nous considérerons plusieurs méthodes proposées pour résoudre les problèmes d'aliassage issus d'une transformation perspective sur les textures planes. Parmi ces méthodes, nous distinguerons essentiellement deux catégories : le filtrage simultané et le filtrage a priori. Nous étudierons les algorithmes existants et nous proposerons de nouvelles méthodes. En particulier, nous proposons un nouvel algorithme utilisant une convolution précise pour l'antialiassage de textures et de contours. Enfin, nous étudierons brièvement l'emploi des textures 3D en synthèse d'images.

V.2 Problèmes de transformation perspective des textures planes

La transformation ou mise en perspective de texture ("texture mapping") a trait au placage de texture sur un objet et à la projection perspective de ce dernier sur l'écran afin d'obtenir l'image finale. Comme nous le verrons ultérieurement, les transformations géométriques comme le placage et la projection perspective provoquent des changements de grille d'échantillonnage qui conduisent à l'aliassage. Dans ce cas, les phénomènes d'aliassage se manifestent notamment sous forme de moirés. L'image V.3.a est un exemple

illustrant la présence de moirés dus à l'application d'une transformation perspective sur la texture originale présentée par l'image V.1. Nous allons étudier l'effet d'une transformation perspective sur la texture initiale dans le domaine spatial et sur son spectre fréquentiel.

V.2.1 Formalisation du problème dans le domaine spatial

Les objets constituant la scène sont définis dans un espace 3D de coordonnées (X,Y,Z) ; la texture est définie dans un espace 2D de coordonnées (u,v) et l'image finale dans l'espace 2D de l'écran de coordonnées (x,y). La figure V.1 est un exemple simple qui montre bien les deux étapes de la mise en perspective : transformations texture-scène (placage) et scène-écran (projection perspective). La composition de ces deux transformations permet de définir la transformation perspective finale (texture-écran). Cette transformation 2D correspond donc à un changement de variables dans le domaine spatial. Signalons que la transformation perspective 2D peut être remplacée par deux transformations monodimensionnelles (cf. paragraphe V.3.2.2).

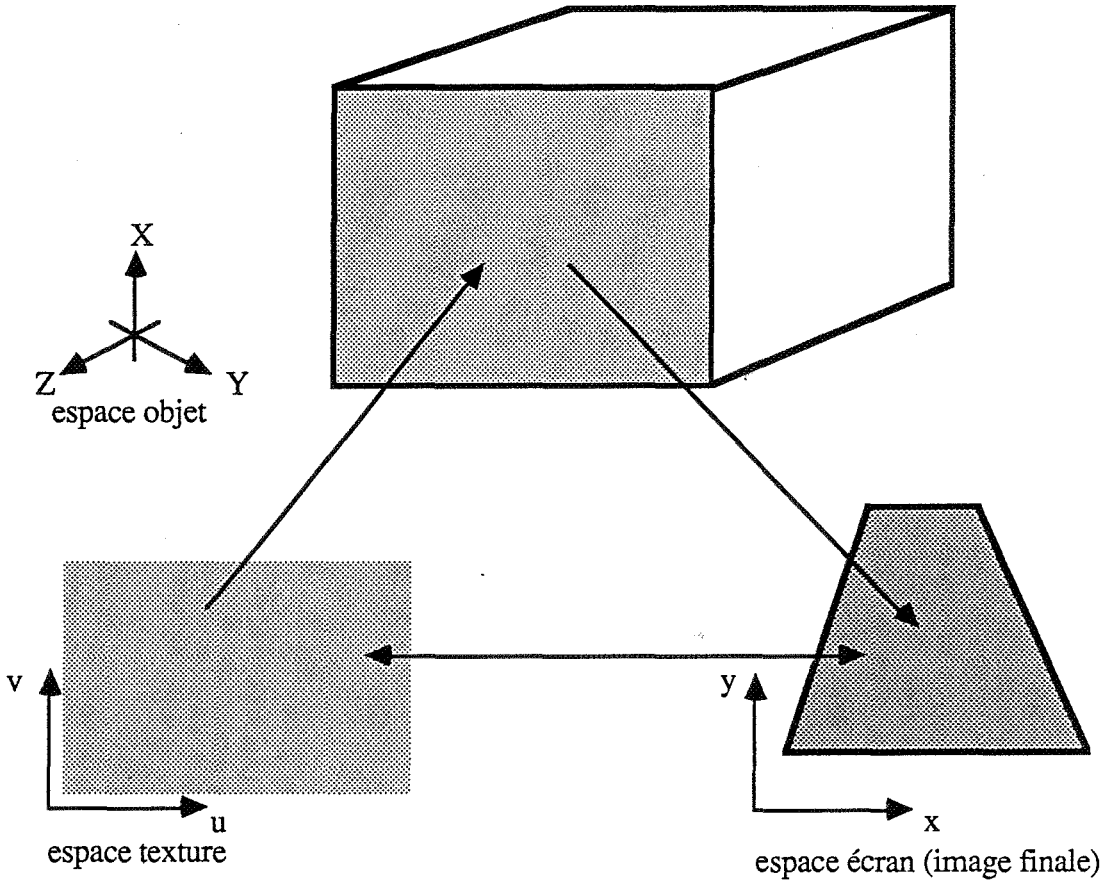


Figure V.1 : Exemple de mise en perspective d'une texture sur une surface plane.

La transformation perspective peut être appliquée directement dans le sens texture-écran ou inversement dans le sens écran-texture. Soit F la transformation perspective directe qui transforme un point (u,v) de la texture initiale en un point (x,y) de l'image finale :

$$F : R^2 \rightarrow R^2$$
$$(u,v) \rightarrow (x,y) = \big(j(u,v),k(u,v)\big) \qquad (V.1)$$

Soient $I_a(x,y)$ le signal analogique représentant l'intensité de l'image finale, et $T_a(u,v)$ le signal analogique représentant l'intensité de la texture originale, sous leur forme continue. L'algorithme suivant décrit la mise en perspective par F de $T_a(u,v)$:

Algorithme V.1, transformation directe :

```
{
 pour v:=0 à n-1 faire
  pour u:=0 à m-1 faire
   {
    x:=j(u,v);
    y:=k(u,v);
    I(x,y):=Ta(u,v);
   }
}
```

La transformation directe est relativement peu utilisée en synthèse d'images. La raison principale est probablement liée au fait que le balayage régulier de l'espace de la texture ne correspond pas forcément dans l'espace de l'écran à un balayage régulier. Le résultat sur l'image finale peut présenter deux types de problèmes :

- des trous éventuels qui correspondent aux pixels non traversés ;
- des pixels coloriés plus d'une fois.

Le premier problème que l'on peut rencontrer également avec certaines transformations géométriques directes comme la rotation classique est généralement inexistant dans une vraie transformation perspective grâce à la compression que la texture subit (cf. paragraphe V.2.1.1). Le deuxième problème n'a pas d'autre conséquence qu'une augmentation des temps de calculs. L'avantage de la transformation directe est l'accès séquentiel à la texture initiale, surtout dans le cas où la texture numérisée stockée ne peut pas être lue que séquentiellement.

La transformation inverse ("inverse mapping") est la méthode la plus courante qui balaye régulièrement l'image finale et transforme chacun de ses pixels en un point de la texture initiale . Nous allons maintenant décrire cette méthode.

Soit F^{-1}, la transformation inverse, qui transforme un point (x,y) de l'image finale en un point (u,v) de la texture originale (cf. figure V.2) :

$$F^{-1} : R^2 \rightarrow R^2$$
$$(x,y) \rightarrow (u,v) = \big(f(x,y),g(x,y)\big) \qquad (V.2)$$

Soient $I_a(x,y)$ le signal analogique représentant l'intensité de l'image finale, et $T_a(u,v)$ le signal analogique représentant l'intensité de la texture originale, sous leur forme continue. Pour tous les points (x,y) où F^{-1} est définie, on pose :

$$I_a(x,y) = T_a(u,v) = T_a\big(f(x,y),g(x,y)\big) \qquad (V.3)$$

L'algorithme suivant décrit la mise en perspective de $T_a(u,v)$ par F^{-1} :

Algorithme V.2, transformation inverse :

```
{
pour y:=0 à n-1 faire
 pour x:=0 à m-1 faire
  {
    u:=f(x,y);
    v:=g(x,y);
    I(x,y):=Ta(u,v);
  }
}
```

Contrairement à la technique précédente, la transformation inverse nécessite en général l'accès aléatoire à la texture originale, ce qui peut poser un problème d'encombrement de la mémoire dans le cas des textures numérisées, en particulier, si plusieurs textures de ce type sont utilisées simultanément.

La relation V.3 représente une transformation inverse (cf. figure V.2). Par exemple, dans le cas courant de la mise en perspective d'une texture sur une surface plane, la transformation inverse F^{-1} est une homographie définie par les relations de la forme :

$$u=f(x,y)= \frac{ax+by+c}{mx+ny+p} \qquad (V.4)$$

$$v=g(x,y)=\frac{dx+ey+f}{mx+ny+p}$$

Avec des surfaces comme les carreaux ("patches") bicubiques, la phase du placage apparaît plus complexe et F^{-1} n'a pas toujours une expression analytique, mais il est toujours possible de calculer $F^{-1}(x,y)$ par la procédure informatique adéquate. En fait, la transformation texture-objet revient essentiellement à une paramétrisation des surfaces de l'objet, qui existe naturellement dans le cas des surfaces paramétrées comme les bicubiques. De plus, pour ce genre de surface, le coût de l'emploi de textures est faible par rapport au coût d'autres calculs nécessaires pour l'affichage de l'objet. C'est pour ces deux raisons que les surfaces comme les carreaux bicubiques portent très souvent des textures.

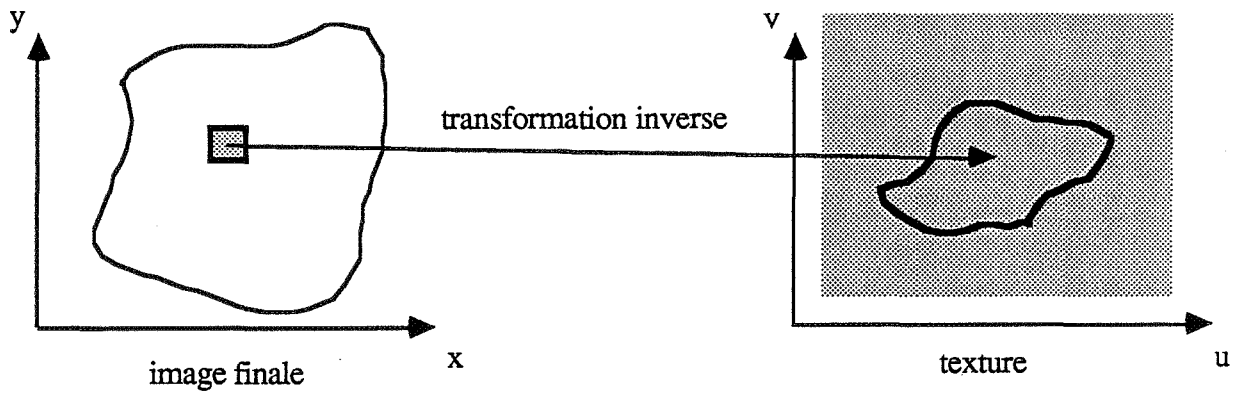


Figure V.2 : Mise en perspective par une transformation inverse.

La relation V.3 définit l'image finale $I_a(x,y)$, sous sa forme analogique. Le problème à résoudre est à nouveau le passage du continu au discret. Comment définir cette image sur la grille régulière d'échantillonnage en évitant l'aliassage ? Le nombre d'échantillons de l'image finale I(i,j), avec i et j entiers, est fixé par la définition de la mémoire de trame. La première solution est donc la limitation du spectre fréquentiel de l'image $I_a(x,y)$ avant son échantillonnage, afin d'écarter les hautes fréquences non représentables (cf. chapitres I et II). Par conséquent, il est indispensable d'effectuer un filtrage passe-bas. Ce filtrage passe-bas est appliqué à $I_a(x,y)$ ou à $T_a(u,v)$ simultanément à la transformation perspective dans le cas des méthodes appelées filtrage simultané, ou directement à $T_a(u,v)$ dans le cas des méthodes appelées filtrage a priori. L'autre solution envisageable mais très peu utilisée est l'échantillonnage stochastique de l'image $I_a(x,y)$ qui transforme l'aliassage en bruit qui est

un défaut moins perceptible (cf. chapitre I). Nous étudierons les différentes méthodes de limitation du spectre fréquentiel ultérieurement.

Un facteur important dans une transformation perspective est la compression de la texture mise en perspective. La compression peut décrire le changement de la grille d'échantillonnage. Par conséquent elle pourra être utilisée ultérieurement comme une mesure locale de l'aliassage induit par la transformation perspective pour certaines méthodes de traitement étudiées dans ce chapitre.

V.2.1.1 Notion de taux de compression

Considérons une transformation perspective dans le cas d'un signal (texture) monodimensionnel (figure V.3) et supposons que le signal analogique $T_a(u)$ puisse être restitué fidèlement à partir des échantillons choisis, $T_e(u)$. On distingue deux zones par rapport au point d'intersection A des axes x et u. Choisissons dx=du et soit $F_x^{-1}(dx) = F^{-1}(x+dx) - F^{-1}(x)$. A gauche du point A, on a $F_x^{-1}(dx)<du$; c'est une zone de dilatation dans laquelle le signal $T_a(F^{-1}(x))$ est sur-échantillonné. Dans cette zone, il suffira de calculer les échantillons de $T_a(F^{-1}(x))$ par une approximation linéaire. A droite du point A, nous avons $F_x^{-1}(dx)>du$; c'est une zone de compression. Dans cette zone, le signal $T_a(F^{-1}(x))$ est sous-échantillonné. Pour le cas monodimensionnel, le filtrage passe-bas antirepliement effectué sur T_a doit donc être localement proportionnel au rapport $\frac{F_x^{-1}(dx)}{du}$ qui définit le taux de compression.

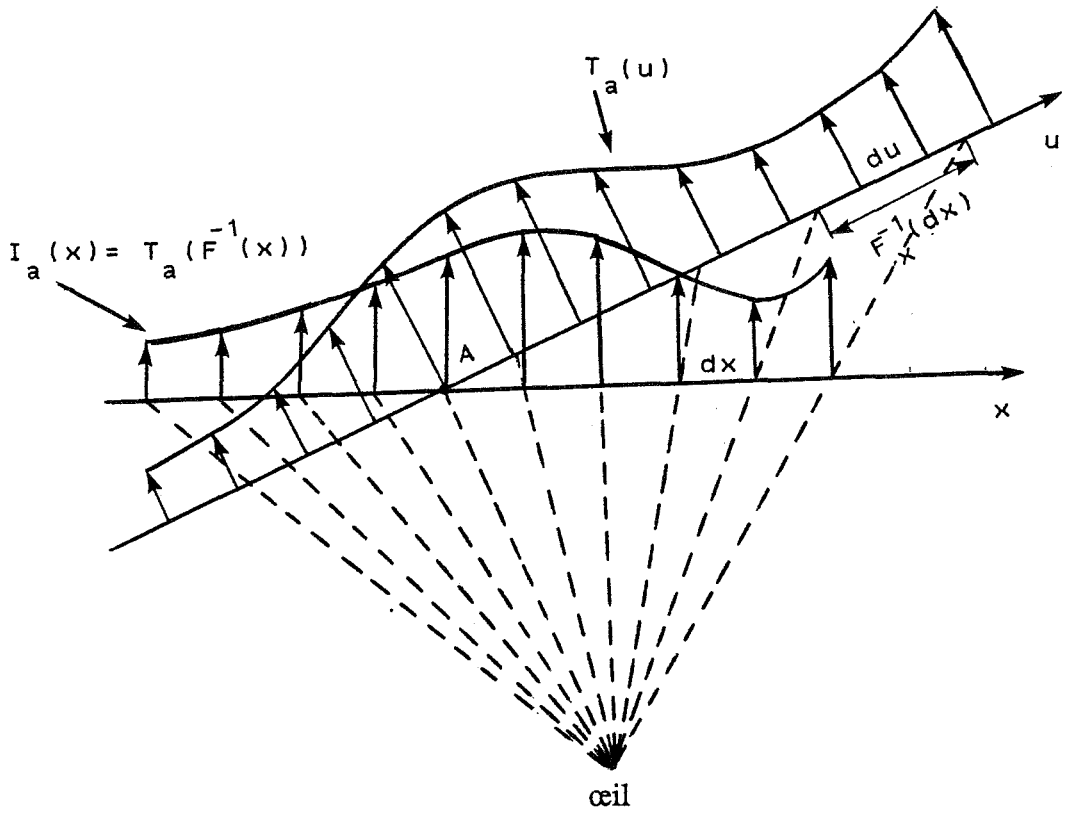


Figure V.3 : Perspective d'un signal (texture) monodimensionnel.

L'argument précédent peut être généralisé au cas de mise en perspective d'un signal bidimensionnel. F^{-1} étant une application de R^2 dans R^2 supposée suffisamment régulière, l'image par F'^{-1}, application linéaire tangente (transformation affine) à F^{-1}, d'un carré de côté un (un pixel par exemple) centré en (x,y) est un parallélogramme (figure V.4). Les longueurs des côtés de ce parallélogramme projetées sur l'axe u (resp. v) sont $|f'_x|$ et $|f'_y|$ (resp. $|g'_x|$ et $|g'_y|$).

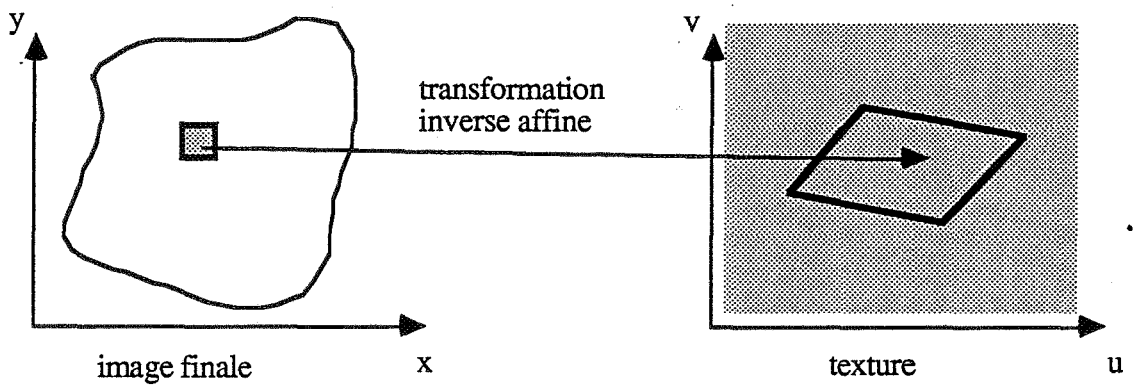


Figure V.4 : Application linéaire tangente à F^{-1}.

Nous définissons les taux de compression en (x,y) par :

$$C_u = (f_x'^2 + g_x'^2)^{1/2}$$
$$C_v = (f_y'^2 + g_y'^2)^{1/2} \qquad (V.5)$$

et le taux unique de compression est défini par :

$$\mathrm{Sup}(C_u, C_v) \qquad (V.6)$$

Nous verrons ultérieurement que l'utilisation d'un taux de compression unique suivant les axes u et v afin de définir un filtrage passe-bas symétrique, pose des problèmes (cf. paragraphe V.3.3.3).

V.2.2 Formalisation du problème dans le domaine fréquentiel

Afin de connaître l'origine du phénomène d'aliassage dû à l'application d'une transformation perspective sur une texture, il est très important d'étudier l'effet de cette transformation sur le spectre fréquentiel de la texture. Pour des raisons de simplicité, nous avons réalisé cette étude dans le cas monodimensionnel [GHAZ 85]. Nous avons montré que dans le cas de la figure V.5, chaque composante de la fréquence f de la texture initiale atteint une fréquence f'(X) au point X après la transformation perspective :

$$f'(X) = \frac{f.[U.\sin(\theta)+d]^2}{d.[e.\sin(\theta)+d.\cos(\theta)]} \qquad (V.7)$$

si θ est nul, alors : $f'(X) = f$;
si θ vaut 90 degrés, alors : $f'(X) = \frac{f.(U+d)^2}{e.d}$; dans ce cas, f'(X) tend vers $+\infty$ lorsque d tend vers 0 ou lorsque e tend vers 0.
pour tout θ, f' tend vers $+\infty$ lorsque d tend vers 0 ou lorsque U tend vers $+\infty$.

Ces résultats sont logiques et confirment la validité de la relation V.7.

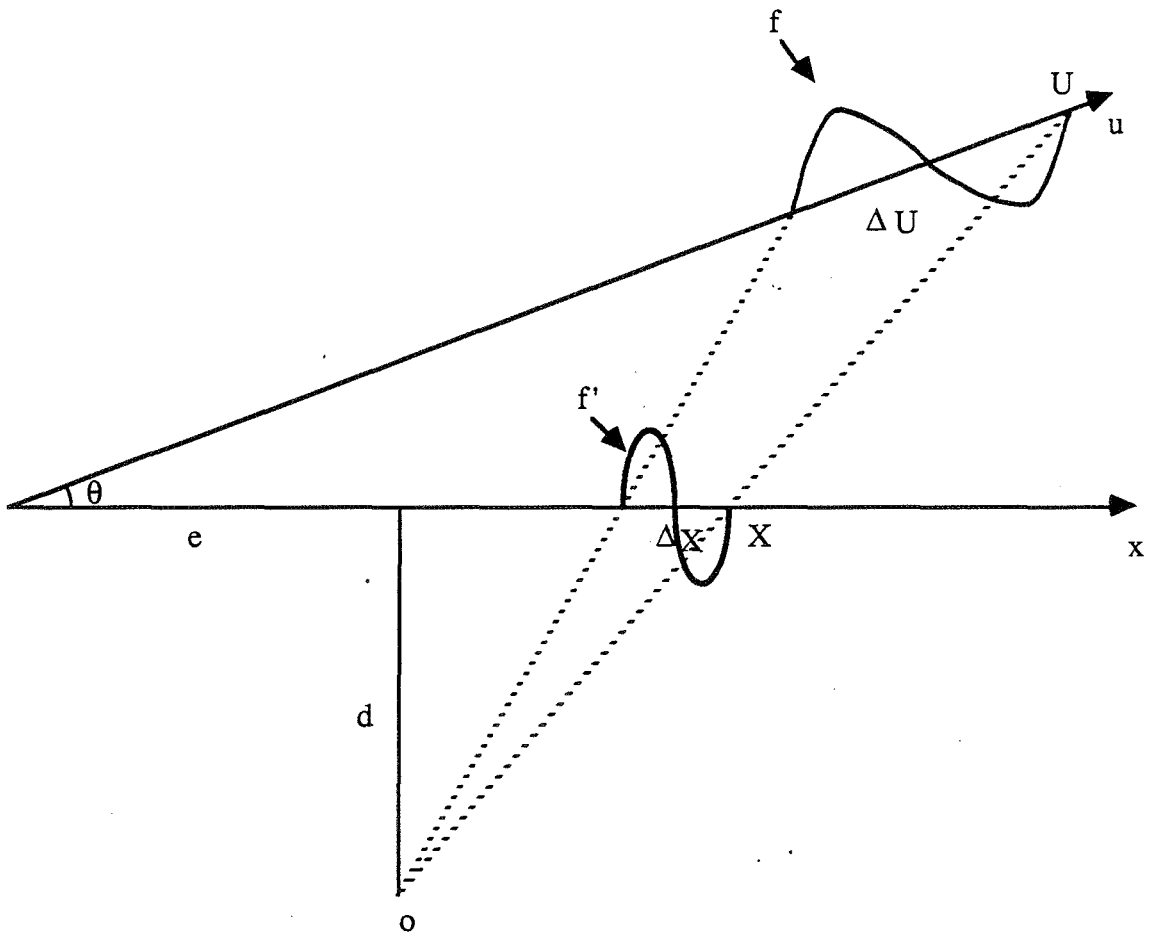


Figure V.5 : Augmentation de la fréquence locale d'une composante due à la transformation perspective.

La relation V.7 met en évidence la modification des fréquences d'un signal monodimensionnel due à l'application d'une transformation perspective. Dans le cas des signaux (textures) définis par une suite de cosinus, cette relation peut être utilisée dans les deux situations suivantes :

1- Le changement de fréquence pour une transformation donnée étant connu, on peut effectuer un traitement d'antialiassage direct. Dans ce cas, il suffit de supprimer les hautes fréquences non représentables de la texture mise en perspective.

2- Pour une transformation donnée, on peut mettre directement en perspective (avec une approximation) la texture originale en changeant la fréquence locale de chacune des composantes de la texture.

V.3 Méthodes de traitement

V.3.1 Introduction

Comme nous l'avons vu précédemment, l'application d'une transformation perspective sur une texture peut produire une compression de texture dans le domaine spatial et, par conséquent, un élargissement des zones hautes fréquences du spectre bidimensionnel de la texture dans le domaine fréquentiel. Il en résulte un phénomène d'aliassage sur l'image finale, principalement sous forme de moirés.

Comme dans les autres cas d'aliassage présentés auparavant, la solution la plus souvent retenue pour éviter ou diminuer sensiblement le phénomène d'aliassage consiste en un préfiltrage passe-bas approprié. Deux approches sont possibles pour effectuer ce préfiltrage dans le domaine spatial :

- les méthodes de filtrage simultané à la transformation perspective (direct ou inverse) essaient d'intégrer numériquement sur la texture initiale l'ensemble des informations relatives à la zone correspondant à un pixel de l'image finale.

- les méthodes de filtrage a priori utilisent une ou plusieurs versions préfiltrées et rééchantillonnées de la texture initiale permettant d'éviter les phénomènes d'aliassage. Le filtrage a priori de la texture permet donc une utilisation directe de la transformation perspective inverse F^{-1} représentée par la relation V.3.

Nous allons étudier chacune de ces deux classes de méthodes du point de vue de leurs qualités et de leurs temps de calcul. Pour les exemples présentés dans ce chapitre nous avons utilisé des textures synthétiques hautes fréquences qui sont les textures les plus difficiles à employer correctement. De plus, ces textures sont structurées parce que le système oeil-cerveau est davantage sensible aux défauts présents dans les images structurées. Les différentes méthodes étudiées dans ce chapitre sont décrites pour des images en niveaux de gris. Pour des images en couleur, il faut appliquer la même technique à chacune des trois composantes rouge, verte et bleue de l'image.

V.3.2 Filtrage simultané

Nous proposons le nom de filtrage simultané pour une classe de méthodes pour lesquelles le calcul de la transformation perspective et le préfiltrage ont lieu en même temps,

contrairement au filtrage a priori où celui-ci précède l'application de la transformation perspective. Parmi les méthodes de cette classe, nous abordons quelques unes des principales déjà publiées. Ensuite, nous étudions une méthode en deux passes et nous proposons un nouvel algorithme d'une grande précision utilisant une transformation perspective directe.

V.3.2.1 Description des méthodes existantes

Une des premières méthodes du type filtrage simultané à la transformation perspective est celle développée par Blinn et Newell [BLIN 76] en 1976. Le filtre utilisé dans [BLIN 76] est une pyramide à base carrée B de taille 2X2 pixels qui est centrée au centre d'un pixel de l'image finale. En approximant localement F^{-1} par son application linéaire tangente, les auteurs calculent l'intensité des points de texture contribuant à B. Ces intensités sont pondérées par les altitudes de la pyramide déformée par F^{-1}. Le filtrage est donc réalisé sur la texture originale.

Feibush, Levoy et Cook [FEIB 80] ont publié une méthode de ce type en 1980. Ils définissent tout d'abord un masque de convolution centré sur le centre de chaque pixel de l'image finale. Supposons, pour simplifier, que le masque choisi ait une base rectangulaire B et que F^{-1} soit donnée par la relation V.4. Les auteurs proposent de calculer $F^{-1}(B)$, qui est dans ce cas un quadrilatère, puis de sélectionner les pixels intérieurs à $F^{-1}(B)$ sur la texture initiale. Après avoir calculé l'image par F de ces pixels, le résultat est convolué numériquement avec le masque. Cette dernière opération est réalisée à l'aide d'une table précalculée donnant les poids du masque pour un sur-échantillonnage régulier de B. Les poids affectés aux valeurs ne figurant pas dans la table sont déterminés par celui du plus proche voisin.

Gangnet, Perny et Coueignoux [GANG 82] ont publié une autre méthode légèrement différente en 1982. Celle-ci se ramène à un sur-échantillonnage localement adaptatif de l'image finale. Le masque de convolution choisi a une base carrée B de côté 1, centrée au centre d'un pixel de l'image finale. Supposons que F^{-1} soit donnée par la relation V.4 ; $F^{-1}(B)$ est alors un quadrilatère. La longueur de la plus grande diagonale de $F^{-1}(B)$ est utilisée comme notion de taux de compression unique local pour déterminer le pas de sur-échantillonnage de B qui est ainsi divisé en plusieurs sous-pixels. On calcule l'image inverse du centre de chaque sous-pixel par F^{-1}. L'intensité en un sous-pixel de la sur-grille régulière est obtenue par une interpolation bilinéaire dans l'image de la texture car l'image inverse du centre d'un sous-pixel ne coïncide pas forcément avec le centre d'un pixel de la texture.

L'approximation bilinéaire est nécessaire dans le cas d'une texture numérique (en général une texture naturelle numérisée). Dans le cas d'une texture synthétique, l'intensité d'un sous-pixel est la valeur calculée pour l'image inverse de son centre par F^{-1}. L'ensemble des intensités obtenues est ensuite filtré par une fonction de pondération qui est un Sinc bidimensionnel pour obtenir l'intensité en un pixel de la grille d'affichage. Sur les images obtenues avec cette méthode, on peut remarquer la persistance d'effets d'aliassage dans les zones très comprimées où le nombre maximal prédéfini de sous-divisions est insuffisant pour un échantillonnage correct. L'algorithme suivant décrit cette méthode.

Algorithme V.3, méthode de [GANG 82] :

```
constante n_max    /* nombre maximal de divisions d'un pixel de l'image finale */
{
  pour chaque pixel (i,j) de l'image finale faire
  {
    calculer les coordonnées des 4 sommets de F-1(B);
    calculer la longueur de la plus grande diagonale, de F-1(B);
    calculer le nombre de divisions, n, de B;
    si n>n_max alors n:=n_max;
    diviser B en n sous-pixels;
    pour chaque sous-pixel (x,y)faire;
    {
      u:=f(x,y);
      v:=g(x,y);
      calculer I(u,v) par 4 lectures et 1 approximation bilinéaire;
    }
    calculer l'intensité finale, I(i,j), par une moyenne pondérée (Sinc);
  }
}
```

La division de B en fonction d'un taux de compression unique peut augmenter sensiblement le volume des calculs nécessaires si les deux taux de compression sont très différents selon les deux directions. L'utilisation de deux taux de compression est donc conseillée.

La méthode de [FEIB 80] somme toute l'information disponible relative à un pixel de l'image finale mais effectue une convolution sur une grille irrégulière alors que la deuxième méthode interpole l'information mais somme sur une grille régulière. D'autre part, le volume des calculs est souvent plus important dans la première méthode.

V.3.2.2 Méthode en deux passes

La transformation perspective, comme d'autres transformations géométriques 2D telle que la rotation, peut être remplacée par deux transformations monodimensionnelles

consécutives [CATM 80] [SMIT 87]. Une méthode de mise en perspective en deux passes décompose donc en deux transformations monodimensionnelles successives la transformation perspective directe F décrite par la relation V.1. La première transformation est horizontale et elle est appliquée à la texture originale (par ligne ou par colonne de balayage) :

$$(x_i,y_i) = (x,v) = \big(j(u,v),v\big) \qquad (V.8)$$

Il en résulte une image intermédiaire qui est représentée par l'image V.2.a, la texture initiale étant celle de l'image V.1. La deuxième transformation, qui est verticale, est appliquée à l'image intermédiaire (par ligne ou par colonne de balayage) afin d'obtenir l'image finale (cf. image V.2.c).

$$(x,y) = \big(x_i,k\big(h(x_i,v),v\big)\big) \qquad (V.9)$$

où $h(x_i,v)$ est la solution de l'équation $x_i=j(u,v)$ pour u.

L'avantage d'une telle méthode réside dans le fait que l'on n'effectue que des transformations et des filtrages passe-bas monodimensionnels (par ligne ou par colonne de balayage) qui peuvent être calculés plus facilement par une technique de type "pipeline". De plus, par sa simplicité, cette méthode permet une implantation matérielle. Les principes de décomposition des transformations en deux passes sont souvent utilisés dans les machines de trucage d'images de télévision en temps réel, grâce à l'implantation matérielle des méthodes à deux passes.

Nous allons présenter une méthode en deux passes inspirée de [CATM 80]. Cette méthode essaie d'éviter les problèmes des hautes fréquences induites par la compression de la texture initiale, grâce à deux filtrage passe-bas monodimensionnels adaptatifs adéquats.

Lors de la phase horizontale de l'algorithme définie par la relation V.8, on calcule le taux local de compression horizontale, C_u, décrit par la relation V.5. La valeur de C_u au pixel (u,v) de la la texture initiale est employée pour définir localement le filtre passe-bas monodimensionnel centré sur ce pixel. Ce filtre normalisé à RIF (cf. chapitre II) utilise une fonction Sinc pour la pondération. De plus, la taille du support de filtre est choisie suffisamment grande afin d'obtenir un filtrage assez sophistiqué. Signalons que l'opération de filtrage est effectuée en utilisant des table de conversion qui minimisent les temps de

calculs pour trouver l'intensité de chaque pixel (x_i,y_i) de l'image intermédiaire. L'image V.2.b montre les résultats obtenus par cette première phase de transformation et de filtrage. L'image V.2.a représente la même transformation sans aucun filtrage.

La deuxième phase définie par la relation V.9 est similaire à l'étape précédente. Le taux local de compression verticale, C_v, permet de définir le filtre passe-bas monodimensionnel local. L'opération de filtrage est identique à celle de la première phase. Les résultats obtenus par cette méthode après la deuxième phase sont montrés sur l'image V.2.d. L'image V.2.c représente les résultats obtenus sans aucun filtrage.

V.3.2.3 Mise en perspective par une transformation directe et une convolution précise

Soit A"B"C"D" l'image inverse d'un pixel d'affichage, ABCD, sur l'espace de la texture initiale (cf. figure V.6) obtenue par une application linéaire tangente à F^{-1} (transformation affine). La plupart des méthodes de mise en perspective de textures essaient d'intégrer numériquement au mieux l'ensemble de l'information de la texture initiale correspondant à A"B"C"D" :

- soit en utilisant une convolution entre A"B"C"D" (ou souvent sa boîte englobante) et une fenêtre de Fourier, ce qui revient à réaliser une moyenne simple, comme dans [WILL 83], [CROW 84], [GLAS 86], [OKA 87], ... ;

- soit en convoluant A"B"C"D" (ou sa boîte englobante) avec une fonction définissant un filtre passe-bas à RIF, ce qui est équivalant à une moyenne pondérée : [GANG 82], [GANG 84], [HECK 86], ... ;

afin de l'attribuer au pixel ABCD.

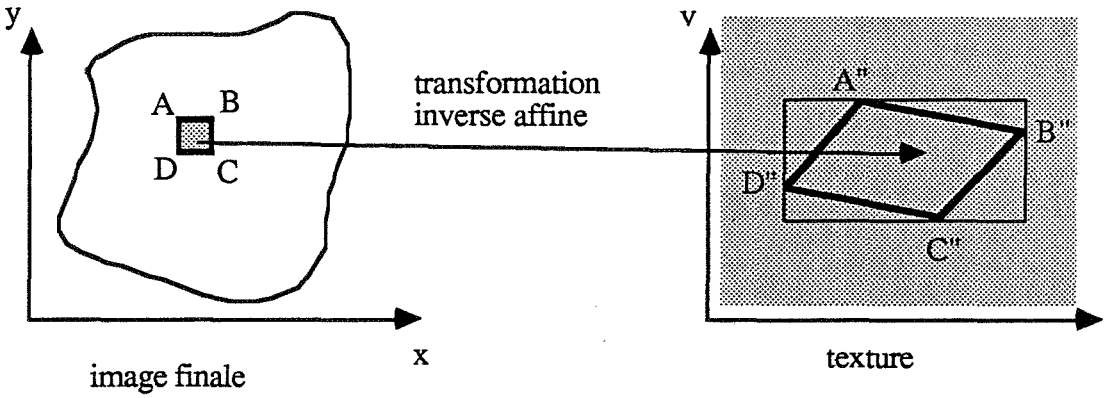


Figure V.6 : Image inverse d'un pixel d'affichage sur l'espace de la texture initiale par l'application linéaire tangente à F^{-1} et sa boîte englobante.

Par exemple, la méthode de Crow [CROW 84], peut donner la moyenne simple des intensités des pixels de la texture initiale qui se trouvent dans la boîte englobante de A"B"C"D" (cf. paragraphe V.3.3.5). La méthode de Glassner [GLAS 86] améliore la qualité des images obtenues avec la méthode précédente en calculant la moyenne simple des intensités des pixels qui sont supposés se trouver, avec un certain degré de précision, à l'intérieur de A"B"C"D" (cf. paragraphe V.3.3.6).

La méthode la plus correcte est en fait celle qui peut intégrer précisément l'ensemble de l'information relative à A'B'C'D', l'image inverse de ABCD par F^{-1} sur la texture initiale (cf. figure V.7). A'B'C'D' peut être une zone délimitée par un contour quelconque dans le cas général. Nous allons présenter dans ce paragraphe, une nouvelle méthode de ce type.

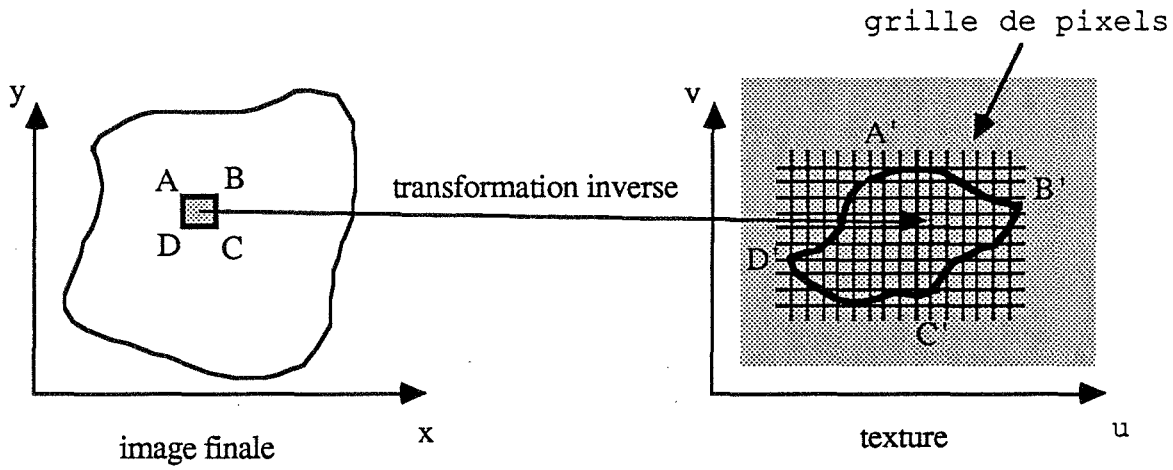


Figure V.7 : Image inverse d'un pixel d'affichage sur la texture initiale par F^{-1}.

Cette nouvelle méthode est simple et générale. Elle peut produire une convolution assez précise avec une fonction de pondération quelconque dans l'espace de l'image finale. De plus, cette méthode peut directement procéder à l'antialiassage des contours de l'image finale. C'est un autre avantage de cette méthode par rapport à la plupart des méthodes existantes.

Nous utilisons la transformation perspective directe F décrite par la relation V.1. L'avantage d'utiliser une telle transformation est la possibilité d'une intégration naturelle de l'ensemble de l'information relative à un pixel de l'image finale. En effet, avec la transformation F, tous les pixels de la texture initiale situés à l'intérieur de A'B'C'D' sont directement plaqués dans ABCD (cf. figure V.8).

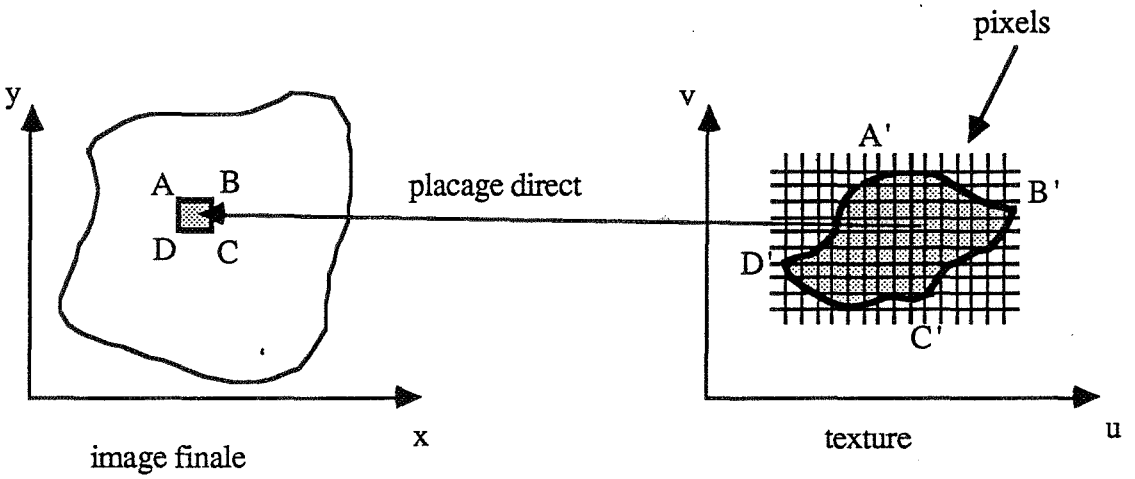


Figure V.8 : Placage direct des pixels de la texture intérieurs à A'B'C'D' en utilisant F.

Nous allons décrire cette méthode dans les deux cas suivants :

-utilisation d'une moyenne simple ;
-utilisation d'une moyenne pondérée.

1- Utilisation d'une moyenne simple :

La convolution de A'B'C'D' avec une fenêtre de Fourier est triviale avec cette méthode. Nous avons uniquement besoin de compter le nombre n de pixels qui sont plaqués dans ABCD et la somme I de leurs intensités. L'intensité du pixel ABCD sera calculée par $\frac{I}{n}$. La moyenne simple des intensités peut être théoriquement considérée comme un filtrage passe-bas non sophistiqué. Avec cette méthode, les images obtenues sont de très bonne qualité. En effet, l'utilisation d'une fenêtre de Fourier est compensée par la précision de la convolution, ce qui n'est pas toujours le cas pour la plupart des autres méthodes.

Nous pouvons améliorer les résultats obtenus avec cette méthode en élargissant la contribution de l'intensité de chaque pixel plaqué aux voisins directs de ABCD. A titre d'exemple, considérons le cas de la figure V.9.a. Soient (i,j) les coordonnées du centre de ABCD, i(u,v) l'intensité d'un pixel plaqué (u,v) en utilisant F : (x,y)=F(u,v), x<i et y>j. Nous avons expérimentalement trouvé que la contribution de i(u,v) dans le calcul de l'intensité des pixels (i-1,j) et (i,j+1) peut améliorer la qualité des images obtenues. De plus, nous pouvons éviter les éventuels problèmes mineurs d'aliassage dans le cas où la compression est faible (quand ABCD a presque la même taille que A'B'C'D').

Le traitement d'antialiassage des contours de l'image finale est un cas simplifié de ce que l'on décrira ci-dessous pour l'emploi d'une moyenne pondérée.

2- Moyenne pondérée :

Cette méthode peut être employée plus efficacement avec un filtrage passe-bas plus sophistiqué utilisant une fonction de pondération quelconque (Sinc, gaussienne, ...) avec une base circulaire. La symétrie de la base circulaire permet de précalculer la fonction de pondération et d'utiliser une table de conversion ("lookup table").

Pour un pixel (u,v) plaqué en utilisant F: (x,y)=F(u,v), nous calculons la distance r entre (x,y) et le centre de chaque pixel approprié situé dans un voisinage déterminé en fonction du rayon R de la base du filtre passe-bas (cf.figure V.9.b). Nous comparons r à R. Si r≤R, alors l'intensité i(u,v) est pondérée avec la valeur de la fonction de pondération au point le plus proche de r et elle contribue à l'intensité du pixel (cf. algorithme V.4).

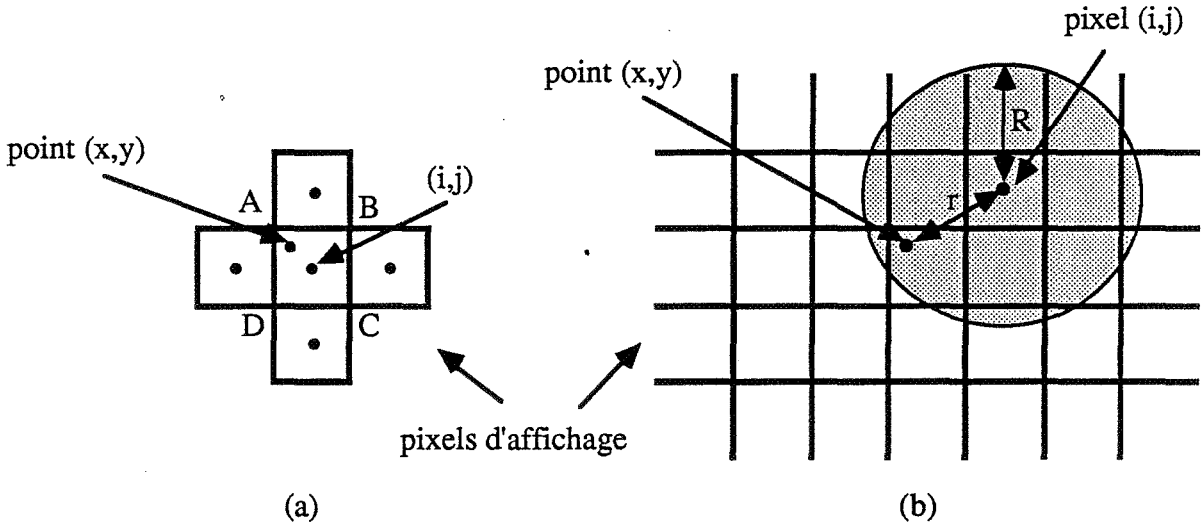


Figure V.9 : Contribution d'un pixel de texture à l'intensité d'un pixel de son voisinage.

Le contour de l'image finale correspond aux frontières de la texture initiale : (u_{min}, v), (u_{max},v) pour $v_{min} \leq v \leq v_{max}$, et (u,$v_{min}$), (u,$v_{max}$) pour $u_{min} < u < u_{max}$. Le cas d'un pixel (i,j) appartenant au contour de l'image finale est représenté par la figure V.10. Une solution simple avec cette méthode pour l'antialiassage du contour est de poursuivre la transformation directe de quelques pas supplémentaires au delà des frontières de la texture initiale (cf. algorithme V.4). Au delà des frontières de la texture, l'intensité de chaque point transformé est celle du fond en (i,j).

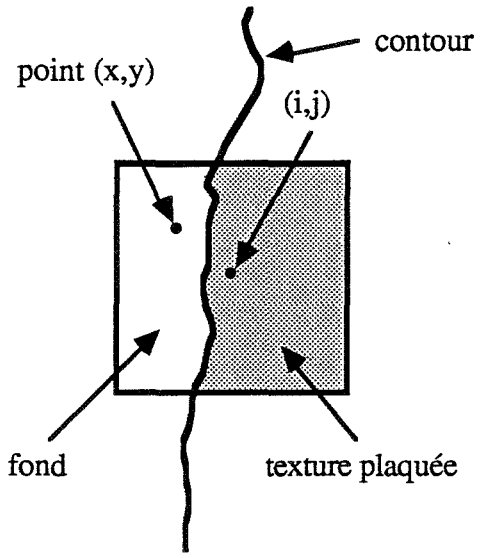


Figure V.10 : Un pixel de contour de l'image finale.

L'algorithme suivant résume les principes de cette méthode quand une moyenne pondérée est utilisée.

Algorithme V.4, Transformation directe et convolution précise :

```
{ /* programme principal */
 /* initialisations */
 pour n=0 à R faire TAB[n]=V(n);      /* V(n): valeur de la pondération en n */
 pour chaque pixel (i,j) de l'image finale faire
  I[j][i]=P[j][i]=0;      /* I[j][i]: intensité du pixel (i,j), P[j][i]: poids du pixel (i,j) */

 / * transformation directe */
 pour v=vmin à vmax faire
  pour u=umin à umax faire
   {
    x=j(u,v);      y=k(u,v);      /* cf. relation V.1 */
    calculer la distance r entre (x,y) et chaque pixel approprié (i,j) de son voisinage;
    si r≤R alors
     {
      I[j][i]=I[j][i]+TAB[round(r)]*i(u,v);  /* i(u,v): l'intensité de texture en (u,v) */
      P[j][i]=P[j][i]+TAB[round(r)];
     }
   }

/* pixels du contour */
pour v=vmin à vmax faire
 {
   Frontière_U(umin,v,-1);
   Frontière_U(umax,v,+1);
 }
pour u=umin+1 à umax-1 faire
 {
   Frontière_V(u,vmin,-1);
   Frontière_V(u,vmax,+1);
 }

/* placage de texture*/
pour chaque pixel (i,j) de l'image finale faire
    afficher (i,j) avec l'intensité I[j][i]/P[j][i];
}


Frontière_U(u,v,n)
{
i=round(j(u,v));  j=round(k(u,v));
u=u+n;    x=j(u,v);      y=k(u,v);
calculer la distance r entre (x,y) et (i,j);
tant que ( r≤R ) faire
   {
    I[j][i]=I[j][i]+TAB[round(r)]*fond(i,j);  /* fond(i,j): intensité du fond en (i,j) */
    P[j][i]=P[j][i]+TAB[round(r)];
    u=u+n;      x=j(u,v);      y=k(u,v);
    calculer la distance r entre (x,y) et (i,j);

   }
}

Frontière_V(u,v,n)
```

```
{
/* similaire à Frontière_U */
}
```

Remarque 1 : *P* peut être utilisé pour la détection des trous éventuels dans l'image finale (cf. paragraphe V.2.1). En effet, la valeur de *P* pour ces pixels est égale à zéro. Pour trouver l'intensité d'un pixel de ce type, il suffit de faire une interpolation bilinéaire entre ses quatre voisins.

Remarque 2: Il est possible d'utiliser une table de conversion pour obtenir la valeur de la distance r. De plus, comme la valeur de l'intensité maximale d'un pixel est 255, les multiplications dans les boucles intérieures (TAB[round(r)]*i(u,v) ou TAB[round(r)]*fond(i,j)) peuvent être remplacées par l'utilisation d'une table de conversion.

Cette méthode est simple et en général beaucoup plus précise que les autres méthodes de mise en perspective de textures. Elle est très efficace même quand on utilise une moyenne simple. Les résultats obtenus avec cette méthode sont illustrés par les images V.3.b et V.5.b. Dans chaque cas, l'image (a) représente la transformation directe sans aucun antialiassage, l'image (c) un zoom x5 de (a) et l'image (d) un zoom x5 de (b). La texture initiale pour l'image V.3 est celle de l'image V.1 et pour l'image V.5 celle de l'image V.4. Ces deux texture sont hautes fréquences et structurées, ce qui représente le type de textures le plus difficile à traiter pour l'antialiassage.

Les temps de calculs pour cette méthode sont indépendants de la définition de l'image finale mais dépendent de celle de la texture initiale. Un petit nombre d'opérations est nécessaire pour cette méthode. En la comparant aux autres méthodes, elle est en général assez rapide. En particulier, quand la taille de l'image finale est considérable, cette méthode devient plus rapide que les méthodes de transformation inverse. Le tableau suivant montre les temps de "C.P.U." pour l'exemple de l'image V.3 :

Méthode	Temps	Pixels de texture	Pixels d'image finale
transformation directe sans antialiassage	t	1572864	indépendant
transformation directe par cette méthode	2t	1572864	indépendant
transformation inverse de [GANG 84]	2.5t	indépendant	518284

Les rapports des temps fournis dans ce tableau peuvent être encore plus favorables pour cette méthode avec une augmentation de la taille de l'image finale.

V.3.3 Filtrage a priori

Le pas d'échantillonnage de la grille d'affichage est fixé. Au lieu d'échantillonner T_a, on échantillonne une image T_f qui est le résultat d'un filtrage passe-bas appliqué à T_a. Ce filtrage passe-bas doit être compatible avec l'effet de la transformation géométrique sur le spectre fréquentiel de T_a. Comme dans le cas des méthodes de filtrage simultané à la transformation perspective, la détermination des fréquences de coupure du filtre passe-bas est effectuée en considérant des taux de compression liés à la transformation géométrique.

En s'appuyant sur les discussions abordées dans le deuxième chapitre, on peut dire qu'un échantillonnage par point utilisant la relation V.3 produira moins d'aliassage s'il est appliqué à un exemplaire de T_a ayant subi un filtrage passe-bas. Le problème est de déterminer précisément le filtre passe-bas à utiliser ; une fréquence de coupure trop basse provoquera du flou, une fréquence de coupure trop élevée des phénomènes d'aliassage. Le filtrage a priori de la texture doit donc être compatible avec la transformation géométrique appliquée.

Une des premières publications portant sur les méthodes de filtrage a priori est celle de Norton, Rockwood et Skomolski [NORT 82] publiée en 1982. Dans cette méthode, la texture initiale est décrite sous forme d'une série de composantes, son développement en série de Fourier. Dans les zones où la texture mise en perspective n'est plus représentable, une valeur moyenne locale de la texture est utilisée.

Nous allons étudier maintenant les différentes méthodes de filtrage a priori ayant généralement des temps de calcul inférieurs à ceux des méthodes de filtrage simultané. Parmi les méthodes de filtrage a priori on peut envisager l'utilisation des versions basses fréquences de l'image de texture. Cette technique intéressante a été proposée par Williams [WILL 83] en 1983. Dans la suite nous examinerons plus particulièrement diverses variantes de cette idée.

V.3.3.1 Filtrage global

La méthode la plus simple mais la moins efficace parmi les méthodes de filtrage a priori est le préfiltrage global de la texture originale. Cette méthode peut être appliquée si la

compression de la texture originale due à la transformation perspective peut être considérée comme constante en tous les points de l'image finale. Dans le cas du filtrage global, on obtient :

$$I(i,j)= T_f(f(i,j),g(i,j)) \qquad (V.10)$$

où T_f est la texture filtrée, la valeur de T_f en (i,j) (centre d'un pixel de l'image finale) par F^{-1} étant obtenue par une approximation bilinéaire. Les fréquences de coupure du filtre passe-bas utilisé sont obtenues à partir des taux de compression. Cette méthode très simple peut être appliquée dans certains cas, en particulier à des facettes de taille raisonnable dont l'angle avec le plan de l'écran est faible. L'application de cette méthode dans le cas général d'une transformation perspective peut poser de nombreux problèmes. En fait, dans le cas où la compression ne peut plus être considérée comme constante, le préfiltrage passe-bas le plus raisonnable est celui qui donne un compromis entre les zones très comprimées et celles faiblement comprimées. Cependant, le résultat de l'application d'une telle méthode n'est pas très acceptable. En général, on constate que l'image finale contient des défauts d'aliassage dans les zones très comprimées où le préfiltrage appliqué est insuffisant. Dans les zones peu comprimées, l'image finale est floue à cause du sur-filtrage.

V.3.3.2 Filtrage adaptatif par découpage

Nous avons proposé dans [GANG 84] une méthode de préfiltrage passe-bas adaptatif pour la mise en perspective d'une texture sur un plan qui convient bien aux systèmes disposant d'une convolution rapide pour une fenêtre rectangulaire de l'image. Dans le cas où F^{-1} est une homographie définie par la relation V.4, on peut tirer profit de ses propriétés projectives pour mettre en place un préfiltrage passe-bas qui est localement proportionnel aux taux de compression. Soit P un polygone à texturer dans l'image finale ; connaissant les images U et V par F des points à l'infini des axes u et v de la texture originale, il est toujours possible de trouver un quadrilatère ABCD contenant P tel que ABCDUV soit un quadrilatère complet du plan projectif [BERG 79] (cf. figure V.11).

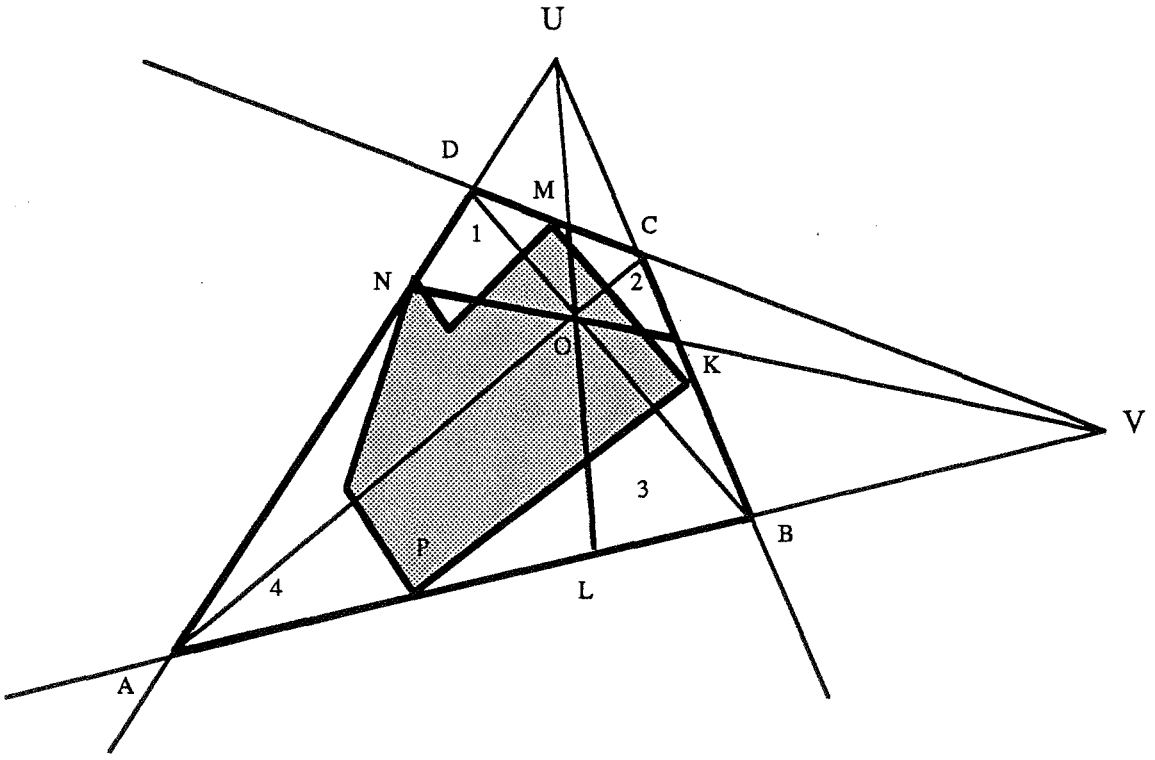


Figure V.11 : Division d'un quadrilatère.

F^{-1} conservant le birapport de quatre points alignés, ADNU est donc une division harmonique ; ceci permet de découper ABCD en quatre quadrilatères : ALON, OMDN, LBKO et OKCM (cf. figure V.11). Ce procédé est récursif. Le test d'arrêt peut être défini par les dimensions des fils ou la proximité de leurs taux de compression à ceux de leur père (les taux de compression d'un quadrilatère sont assimilés à ceux de son centre).

L'image par F^{-1} de chacun des quadrilatères dans l'image de la texture originale est un rectangle dont les bords sont parallèles aux axes u et v ; c'est une propriété intéressante qui facilite le filtrage passe-bas. Chaque quadrilatère est alors rempli par une approximation bilinéaire effectuée sur le rectangle correspondant filtré. On ne traite évidemment que les pixels intérieurs à P.

Algorithme V.4, filtrage adaptatif par découpage :

```
{ /* programme principal */
  calculer les coordonnées de U et V;
  calculer les coordonnées de A,B,C et D;
   T0:=0;
   découpage(T0, coordonnées de U,V,A,B,C,D);
}

découpage(réels : T0, coordonnées de U,V,A,B,C,D)
constantes  S0,ε
{
   calculer xO et yO   /* O est le centre de ABCD */;
   calculer Cu(xO,yO) et Cv(xO,yO);  /* cf. relation V.5 */
   taux:= Sup(Cu(xO,yO),Cv(xO,yO));
   calculer SABCD;
   si (non(T0-ε<taux<T0+ε)) et SABCD>S0 alors
   {
     calculer les coordonnées de N,M,K et L;
     découpage(taux, coordonnées de U,V,A,L,O,N);
     découpage(taux, coordonnées de U,V,L,B,K,O);
     découpage(taux, coordonnées de U,V,O,K,C,M);
     découpage(taux, coordonnées de U,V,O,M,D,N);
   }
  sinon filtrage(Cu(xO,yO),Cv(xO,yO), coordonnées de A,B,C,D)
}

filtrage(réels :Cu,Cv, coordonnées de A,B,C,D)
{
   calculer les coordonnées de F-1(A),F-1(B),F-1(C),F-1(D);
   calculer les fréquences de coupure en fonction de Cu et Cv;
   filtrer le rectangle correspondant au quadrilatère ABCD;  /* cf. Algorithme II.1 */
 }
```

L'application de cette méthode sur des machines qui offrent la possibilité d'une convolution instantanée est particulièrement intéressante. Cependant, le découpage est basé sur les propriétés projectives des plans et cette méthode ne peut pas être utilisée pour des surfaces gauches non-facettisées.

V.3.3.3 Filtrage symétrique

Une méthode très intéressante de filtrage a priori qui sert de base pour beaucoup d'autres a été exposée par Williams en 1983 [WILL 83]. Cette méthode utilise une structure pyramidale composée d'un ensemble d'images sources qui sont les résultats de préfiltrages passe-bas successifs sur la texture originale.

Soit $2^n x 2^n$ la dimension de la texture initiale ; une suite d'images T_p de dimensions $2^p x 2^p$, p variant de n à 0, est générée (cf. figure V.12.a). L'intensité de chaque pixel de T_{p-1} étant la moyenne simple de 2x2=4 pixels correspondants de T_p. Une telle structure est appelée "mip map" par l'auteur, signifiant " beaucoup de choses dans une petite place". Nous avons montré dans [GANG 84] que l'utilisation d'une moyenne simple n'est pas suffisante pour que la texture originale soit bien représentée (sans défaut d'aliassage) sur toutes les images de T_p.

Les images de T_p correspondent à des taux de compression de 2^{n-p}, égaux en u et v. Cette structure peut être considérée comme une structure pyramidale (cf. figure V.12.b) où la base de la pyramide est l'image de dimension $2^n x 2^n$ et la coordonnée verticale le taux de compression unique, C, décrit par la relation V.6.

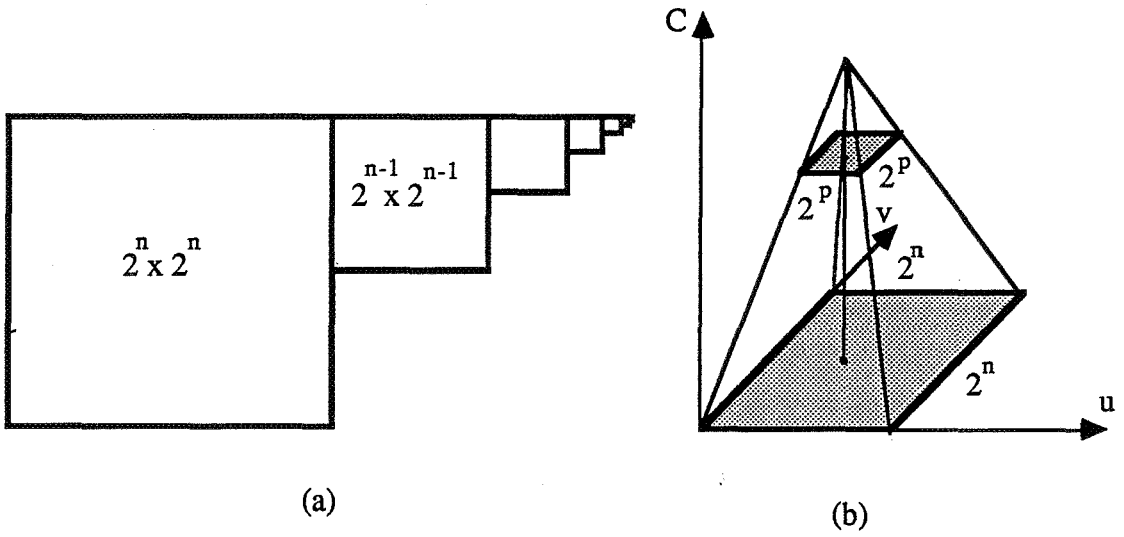


Figure V.12 : Suite T_p ("mip map"), $0 \leq p \leq 8$ et structure pyramidale.

Soit (i,j) le centre d'un pixel de l'image finale ; une fois calculé le taux de compression C en l'image d'un point (i,j) par F^{-1}, on détermine p tel que :

$$2^{n-p-1} \leq C < 2^{n-p} \qquad (V.11)$$

Pour obtenir l'intensité finale I(i,j) du pixel (i,j), on calcule les intensités correspondant à $F^{-1}(i,j)$ sur T_{p-1} et T_p, notées $I_{p-1}(F^{-1}(i,j))$ et $I_p(F^{-1}(i,j))$, à l'aide de huit lectures de pixels et deux approximations bilinéaires. La valeur de I(i,j) est obtenue par une approximation linéaire entre $I_{p-1}(F^{-1}(i,j))$ et $I_p(F^{-1}(i,j))$. L'interpolation bilinéaire des pixels de bord d'une

image T_p est réalisée avec l'autre bord sur la même image dans le cas des textures périodiques. Pour des textures non périodiques, on procède à un fenêtrage des coordonnées u et v pour éviter d'interpoler avec des composants incorrects. L'algorithme suivant décrit cette méthode.

Algorithme V.5, filtrage symétrique :

```
{
 pour chaque pixel (i,j) de l'image finale faire
 {
   u:=f(i,j);
   v:= g(i,j);
   calculer C par la relation V.6;          /* C : taux de compression unique en u et v */
   déterminer p tel que 2^(n-p-1)≤C<2^(n-p);
   calculer I_(p-1)(u,v) par 4 lectures et 1 approximation bilinéaire sur T_(p-1);
   calculer I_p(u,v) par 4 lectures et 1 approximation bilinéaire sur T_p;
   calculer I(i,j) par 1 approximation linéaire entre I_(p-1)(u,v) et I_p(u,v);
 }
}
```

L'intérêt évident de cette méthode est que la texture est filtrée une fois pour toutes et que la représentation pyramidale peut être utilisée quelle que soit la transformation géométrique appliquée. Ceci est plus particulièrement intéressant dans le cas où la même texture est utilisée par de nombreuses transformations géométriques. L'autre avantage de cette méthode est sa généralité ; en fait, elle peut être appliquée aussi bien aux surfaces gauches qu'aux plans. Comme l'auteur le remarque, l'utilisation d'une moyenne simple est une utilisation minimale du filtrage passe-bas et elle peut poser des problèmes d'aliassage. Le problème important de cette méthode est le filtrage symétrique et l'utilisation d'un taux de compression unique dans les deux directions u et v. Dans certains cas, les taux de compression peuvent être très différents selon ces deux directions (cf. image V.6.a). La valeur de C choisie étant le maximum des deux taux de compression C_u et C_v, le filtrage symétrique est alors source de problèmes si C_u est très différent de C_v. Dans ce cas, l'image finale sera très floue à cause d'un sur-filtrage dans le sens où le taux de compression est le plus faible (cf. image V.6.b). La méthode qui suit, est une généralisation de la technique pyramidale et peut éviter les problèmes de cette dernière.

V.3.3.4 Filtrage asymétrique

Nous avons proposé dans [GANG 84] une méthode appelée filtrage asymétrique généralisant la méthode du filtrage symétrique. Cette méthode remplace le taux de compression unique par deux taux de compression distincts et utilise des préfiltrages passe-

bas monodimensionnels perfectionnés. Les images obtenues sont de meilleure qualité que celles réalisées avec la méthode précédente.

Soit $T_{n,n}$ la texture originale de dimension $2^n x 2^n$; nous proposons de construire une suite d'images $T_{p,q}$ de dimensions $2^p x 2^q$, p variant de n à 0, q variant de n à 0, l'image $T_{p,q}$ étant obtenue soit par "sous-échantillonnage" de l'image $T_{p+1,q}$ filtrée suivant u, soit par "sous-échantillonnage" de l'image $T_{p,q+1}$ filtrée suivant v (cf. figure V.13). Le préfiltrage appliqué est un filtrage passe-bas (à RIF) monodimensionnel sophistiqué utilisant une fonction de pondération et un masque d'une longueur importante. Algorithmiquement, ces images sont générées en deux passes monodimensionnelles : une passe horizontale de construction d'une suite d'images $T_{p,n}$, $0 \leq p \leq n$, suivie d'une passe verticale qui, pour chaque $T_{p,n}$ calculée précédemment, génère la suite $T_{p,q}$, $0 \leq q \leq n$. On remarque que les images carrées de la structure pyramidale de la méthode précédente sont remplacées par les images rectangulaires de la suite $T_{p,q}$; sur sa diagonale, cette structure est équivalente à la structure pyramidale (cf. figure V.13). Une telle structure va permettre d'utiliser les taux de compression, C_u et C_v, calculés indépendamment dans les deux directions par la relation V.5. De plus,l'utilisation d'un filtrage sophistiqué résout les problèmes constatés sur le "mip map" généré par la méthode précédente.

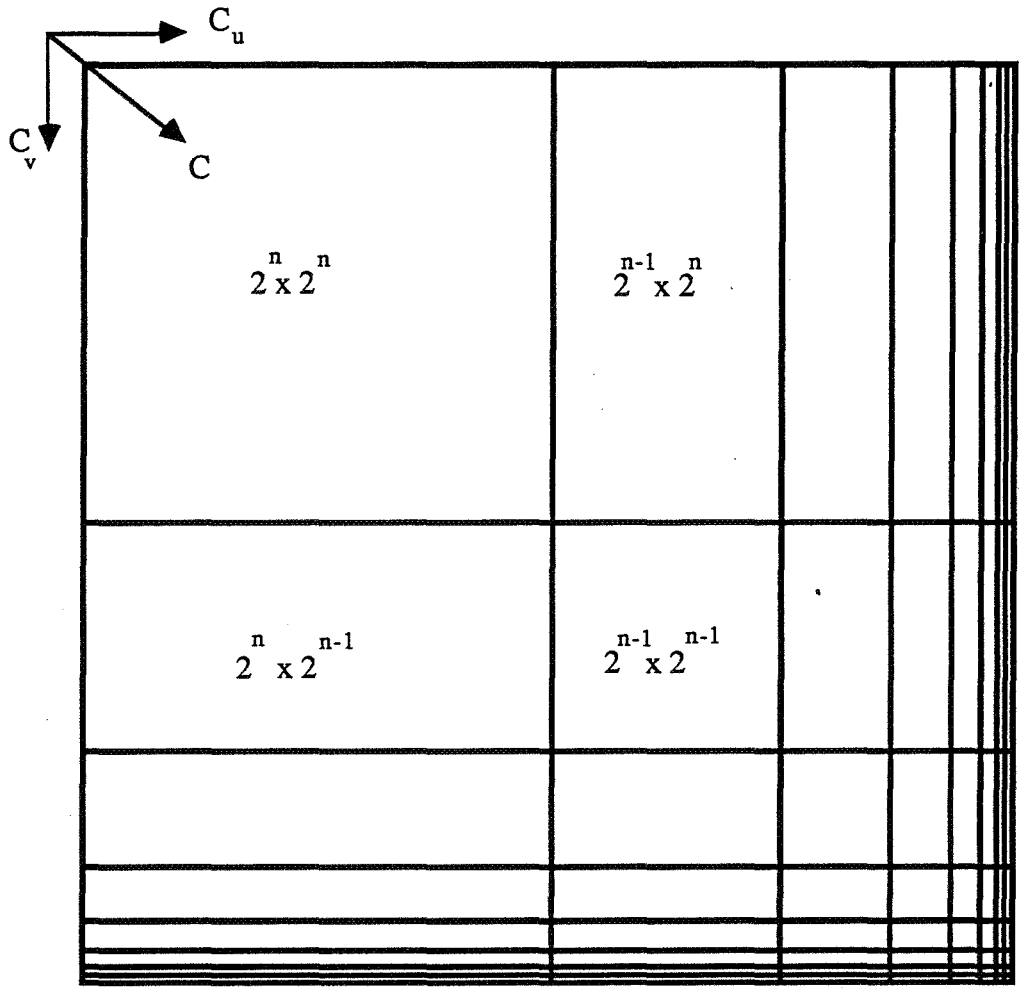


Figure V.13 : Une suite d'images $T_{p,q}$ ("mip map" 2D), $0 \leq p \leq n$, $0 \leq q \leq n$.

Les principes de cette méthode sont les suivants : si i et j sont les coordonnées du centre d'un pixel de l'image finale, on calcule C_u, C_v en $F^{-1}(i,j)$. On détermine p et q de telle sorte que :

$$2^{n-p-1} \leq C_u < 2^{n-p}$$
$$2^{n-q-1} \leq C_u < 2^{n-q} \qquad \text{(V.12)}$$

On calcule les intensités $I_{p-1,q-1}(F^{-1}(i,j))$, $I_{p,q-1}(F^{-1}(i,j))$, $I_{p-1,q}(F^{-1}(i,j))$ et $I_{p,q}(F^{-1}(i,j))$ correspondant à $F^{-1}(i,j)$ sur les images $T_{p-1,q-1}$, $T_{p,q-1}$, $T_{p-1,q}$, $T_{p,q}$ par seize lectures de pixel et par quatre approximations bilinéaires. La valeur de l'intensité finale I(i,j) du pixel (i,j) de l'image finale est obtenue à l'aide d'une approximation bilinéaire entre les quatre intensités que l'on vient de calculer. L'algorithme qui suit illustre cette méthode.

Algorithme V.6, filtrage asymétrique :

{
pour chaque pixel (i,j) de l'image finale **faire**
{
u:=f(i,j);
v:=g(i,j);
calculer C_u par la relation V.5;
calculer C_v par la relation V.5;
déterminer p tel que $2^{n-p-1} \leq C_u < 2^{n-p}$;
déterminer q tel que $2^{n-q-1} \leq C_v < 2^{n-q}$;
calculer $I_{p-1,q-1}(u,v)$ par 4 lectures et 1 approximation bilinéaire sur $T_{p-1,q-1}$;
calculer $I_{p,q-1}(u,v)$ par 4 lectures et 1 approximation bilinéaire sur $T_{p,q-1}$;
calculer $I_{p-1,q}(u,v)$ par 4 lectures et 1 approximation bilinéaire sur $T_{p-1,q}$;
calculer $I_{p,q}(u,v)$ par 4 lectures et 1 approximation bilinéaire sur $T_{p,q}$;
calculer I(i,j) par 1 approximation bilinéaire entre les 4 intensités calculées;
}
}

Comme la précédente, cette méthode est intéressante car la texture est filtrée une fois pour toutes et les différentes vignettes préfiltrées seront utilisées quelle que soit la transformation géométrique appliquée. De plus, l'utilisation d'un filtrage passe-bas plus perfectionné que des moyennes simples permet de résoudre les problèmes d'aliassage visibles sur les vignettes préfiltrées avec la méthode précédente. Comparons l'image V.6.c, générée par la méthode du filtrage asymétrique, avec l'image V.6.b générée par la méthode du filtrage symétrique. Dans les deux cas, la même transformation perspective est appliquée et les taux de compression sont très différents selon les deux directions. Notre méthode de filtrage asymétrique donne de meilleurs résultats ; il n'y a pas de flou dû à un sur-filtrage selon une direction comme avec la méthode précédente grâce à un préfiltrage adapté. De plus, les défauts d'aliassage sont moins visibles que sur les images obtenues avec la méthode précédente grâce à un meilleur filtrage. Comme la précédente, la méthode de préfiltrage asymétrique est générale et elle peut être utilisée également pour des surfaces gauches. Les images V.7.c et V.8.c montrent les résultats obtenus pour un carreau bilinéaire et pour une sphère et les images V.7.d et V.8.d sont des zoom x5 sur les images précédentes. Les images V.7.a et V.8.a sont non traitées et les images V.7.b et V.8.b représentent leur zoom x5.

V.3.3.5 La méthode de cumulation d'intensités

Parmi les autres méthodes de filtrage a priori, on peut signaler la méthode de Crow publiée en 1984 [CROW 84]. Les principes de cette méthode sont très proches de ceux de la

méthode de filtrage asymétrique décrite auparavant. En fait, les différentes vignettes rectangulaires préfiltrées de la texture sont remplacées par une table qui donne la somme des intensités des pixels d'un rectangle quelconque sur la texture originale. Cette valeur d'intensité totale s'obtient par une addition et deux soustractions des valeurs de la table. L'intensité moyenne du rectangle est égale à la valeur d'intensité totale divisée par l'aire (en pixels) du rectangle ; le préfiltrage de la texture originale est donc équivalent à une moyenne simple comme pour la méthode de filtrage symétrique. L'auteur remarque que si on peut étendre cette méthode par l'utilisation d'un filtrage plus précis que la moyenne simple, on obtiendra de meilleurs résultats. On peut résumer cette méthode comme une méthode de filtrage asymétrique dont les préfiltrages utilisés sont les moyennes simples.

V.3.3.6 La méthode avec précision adaptative

En 1986, Glassner a présenté une méthode [GLAS 86] avec précision adaptative qui est une amélioration de la technique précédente. Rappelons qu'un des problèmes de cette dernière est l'intégration de l'information relative à la texture initiale dans la boîte englobante de l'image inverse par l'application linéaire tangente à F^{-1} d'un pixel de l'image finale (cf. figure V.6). Signalons que c'est également le cas pour la méthode de [GANG 84] et le problème est plus important encore pour la méthode de [WILL 83] qui utilise un carré englobant. En utilisant un tableau supplémentaire contenant les variations locales de la texture, la méthode de [GLAS 86] définit le degré de précision de l'intégration de l'information par une moyenne simple sur la texture initiale pour chaque pixel de l'image finale. Glassner divise la surface de la boîte englobante extérieure à A"B"C"D" en rectangles et triangles en fonction du degré de précision (cf. figure V.14). Il enlève l'information relative à chaque rectangle de l'information globale relative à la boîte englobante. Dans les exemples de temps de calcul présentés par l'auteur, cette méthode est environ une fois et demi plus lente que la précédente.

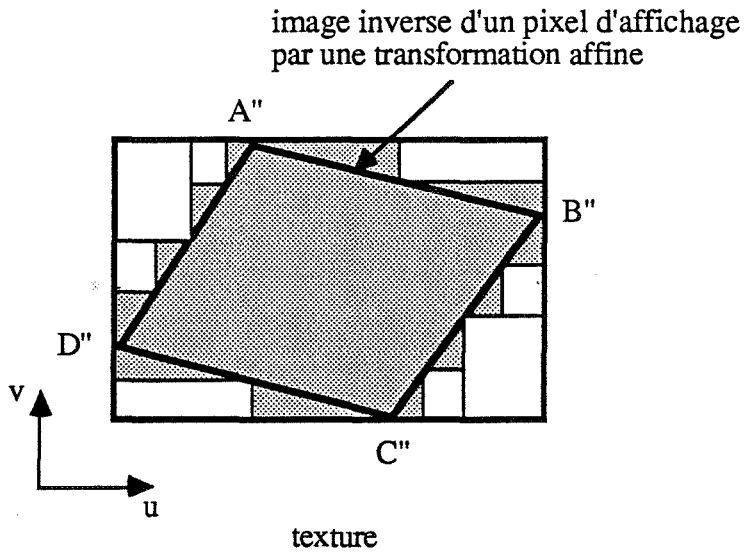


Figure V.14 : Précision adaptative pour l'intégration de l'information.

V.3.3.7 Filtrage par intégrations répétées

La méthode de Heckbert présentée en 1986 [HECK 86] est une autre généralisation de la méthode de Crow [CROW 84]. Il est bien connu en traitement du signal que la convolution d'un rectangle avec lui-même produit un triangle. Rappelons qu'une convolution avec un rectangle correspond à une moyenne simple. Si on continue la convolution par le même rectangle, les fonctions de degré supérieur peuvent être obtenues. On peut obtenir un filtrage passe-bas équivalant à celui obtenu par une fonction polynomiale de degré n-1 avec n convolutions entre un signal et un rectangle. La méthode d'intégrations répétées utilise cette propriété pour obtenir des filtrages passe-bas de bonne qualité.

V.3.4 Conclusion

La plupart des méthodes de filtrage simultané peuvent être très coûteuses en temps de calcul puisqu'il est nécessaire de calculer les images par F ou par F^{-1} en de nombreux points. Il est cependant possible d'optimiser le calcul des relations de type V.4 par un calcul incrémental ne demandant que des additions et deux divisions par point. Parmi les méthodes de cette classe, la technique de convolution précise par une transformation directe produit des images de haute qualité. De plus, les temps de calcul de cette méthode sont raisonnables même en les comparant aux méthodes de filtrage a priori ayant en général des temps de calcul moins importants que le filtrage simultané.

Dans les méthodes utilisant un filtrage a priori, le coût du préfiltrage est fixe et indépendant de l'image finale. Ces méthodes deviennent avantageuse si la même texture est utilisée pour plusieurs objets de la scène afin d'amortir le coût du filtrage a priori. Parmi ces méthodes, celle de Williams [WILL 83] est la plus rapide ; à cause d'un sur-filtrage dans certaines zones, l'image finale peut devenir floue. Notre méthode [GANG 84] grâce à l'utilisation de deux taux de compression et deux préfiltrages monodimensionnels sophistiqués, et la méthode de Glassner [GLAS 86] grâce à sa précision d'intégration de l'information sur la texture initiale peuvent produire des images de meilleure qualité.

Une autre catégorie de méthodes envisageables est celle de l'échantillonnage stochastique [MITC 87]. Comme nous l'avons signalé auparavant, cette technique remplace les défauts d'aliassage par un autre défaut moins visible : du bruit. Les temps de calculs sont importants, en particulier à cause du sur-échantillonnage qui est généralement indispensable. Etant donnés les bonnes performances et le rapport qualité/temps de calculs des méthode décrites dans ce chapitre, l'utilisation de l'échantillonnage stochastique n'est pas intéressante dans ce cas.

Signalons enfin que pour éviter un très grand volume de calculs, il est important d'optimiser au maximum les calculs nécessaires pour les transformations F ou F^{-1} (par exemple, un calcul incrémental dans le cas de la relation V.4).

V.4 Textures 3D

Dans les premières utilisations de textures pour les images de synthèse [CATM 74], ce sont des textures 2D existantes (images) qui ont été numérisées et plaquées sur des surfaces. Le problème de la génération des textures synthétiques se pose depuis leur première utilisation dans [BLIN 76] et le sujet a été étudié abondamment pour les textures 2D et plus récemment étendu au cas 3D [PEAC 85] [PERL 85] [PERL 89].

Des résultats intéressants peuvent être obtenus par diverses méthodes de génération de textures 2D parmi lesquelles on peut citer :

- [MAND 82],[MILL 86], ... pour les méthodes fractales ;
- [SCHA 80],[GHAZ 85], [MAST 87], ... pour les méthodes spectrales ;
- [JAYA 79],[SMIT 84], ... pour les méthodes structurelles ;

- [MONN 81],[GAGA 83],[CARE 87], ... pour les méthodes stochastiques.

cependant, aucune d'entre elles ne permet de résoudre tous les cas envisagés. En fait, la recherche d'un modèle mathématique général pour les textures procédurales 2D ou 3D reste un sujet difficile à maîtriser car il existe de nombreux paramètres subjectifs ou psycho-visuels tels que la forme de l'objet texturé ou le contexte de la texture qui jouent un rôle important dans le réalisme obtenu.

Les texture 3D sont définies pour tous les point de l'espace 3D et les objets pourront être sculptés dans cet espace. Il existe certaines similitudes de principes de génération entre les texture 2D et 3D. Cependant, la génération de textures 3D demeure plus complexe.

La génération d'une texture 3D peut être considérée comme une extension naturelle d'une texture 2D définie par une fonction f(u,v). Chaque point sur chacune des surfaces d'un objet d'une scène 3D a des coordonnées uniques (X,Y,Z) dans l'espace d'objet ; une fonction f(X,Y,Z) peut donc être utilisée pour définir la texture d'un objet. Il est évident que dans ce cas la phase de placage nécessaire pour les textures 2D est inexistante. Cependant, pour une transformation quelconque entre l'espace de la texture et l'espace de l'objet, chaque point (X,Y,Z) de la surface peut être transformé avant l'utilisation de la texture. L'intérêt principal des textures 3D réside dans la continuité de celles-ci sur des surfaces adjacentes, ce qui est généralement difficile à obtenir avec une texture 2D.

Pour clore ce chapitre sur les textures et les moirés, nous allons décrire brièvement la génération et l'antialiassage des textures 3D.

V.4.1 Méthodes de génération

La numérisation des volumes qui est une extension de la numérisation des textures 2D peut être considérée comme une technique d'acquisition de textures 3D. Cette technique est évidemment beaucoup plus compliquée et beaucoup plus coûteuse que dans le cas 2D, car la numérisation 3D est complexe et le stockage des échantillons qui sont les pixels volumiques ("voxel") nécessite une quantité importante de mémoire.

La projection d'une texture 2D numérisée ou synthétique peut être utilisée pour la génération d'une texture 3D. La projection parallèle suivant la direction perpendiculaire au plan de la texture 2D est la méthode la plus connue ; d'ailleurs, elle est souvent utilisée dans la phase du placage d'une texture 2D sur une surface gauche. D'autres projections suivant

une direction quelconque ou l'évolution du plan de la texture 2D autour d'un axe peuvent être utilisées pour obtenir une texture 3D.

Une extension des méthodes spectrales pour la synthèse de textures 2D peut être utilisée pour la génération de textures 3D. Dans ce cas, la texture est définie par une série de cosinus. Les méthodes de [GARD 84] et [GARD 85] utilisent un tel principe. Comme dans le cas 2D, ce modèle spectral est particulièrement bien adapté aux textures de type vagues de la mer, qui ont un aspect plus au moins périodique.

Une autre extension des méthodes 2D pour la génération de textures 3D est celle des méthodes stochastiques. Ces méthodes utilisent souvent une fonction appelée bruit qui représente des valeurs pseudo-aléatoires pour chaque point de l'espace de la texture 3D. Une autre fonction aléatoire utilisée est celle appelée turbulence définie dans [PERL 85]. Ces méthodes sont souvent utilisées pour la synthèse du marbre.

V.4.2 Méthodes d'antialiassage

L'aliassage induit lors de la mise en perspective de textures 2D dépend d'une part du placage de la texture sur l'objet et d'autre part de la projection perspective de l'objet sur l'écran d'affichage. Dans le cas de textures 3D, le placage étant inexistant, l'antialiassage peut donc être effectué a priori indépendamment de la surface qui porte la texture.

L'antialiassage de textures 3D peut être considéré comme une opération de filtrage passe-bas 3D qui est une extension du cas 2D (cf. chapitre II). Dans le domaine spatial, cette opération de filtrage est un produit de convolution tridimensionnel entre la fonction f(X,Y,Z) définissant la texture et la fonction h(X,Y,Z) définissant le filtre passe-bas 3D à RIF. Dans la plupart des cas, la mise en oeuvre directe d'une tel filtrage est assez compliquée. En tout cas, quelle que soit la méthode de filtrage utilisée, le gain de temps est considérable s'il a lieu dans le volume englobant la surface de l'objet.

Comme pour les textures 2D, on peut envisager différentes méthodes d'antialiassage adaptées à chaque cas et à chaque procédure de génération de textures 3D. Les méthodes d'antialiassage utilisées sont essentiellement des extensions de celles utilisées pour la mise en perspective de textures planes.

La première méthode envisageable est évidement le sur-échantillonnage qui peut être appliqué de façon régulière ou à une définition moins importante en utilisant la technique

stochastique (cf. chapitre I) bien adaptée aux textures 3D. La méthode de perturbation de la grille d'échantillonnage 2D (cf. chapitre II) peut être étendue au cas 3D pour effectuer le sur-échantillonnage stochastique. Dans le cas 2D, la perturbation de la grille d'échantillonnage revient à remplacer tout point d'échantillonnage qui est au centre d'un pixel par un point aléatoirement choisi à l'intérieur du pixel. Pour le cas 3D, l'application de cette technique remplace tout point d'échantillonnage régulier au centre d'un voxel par un point aléatoirement sélectionné dans le volume du voxel.

Les méthodes de filtrages passe-bas a priori utilisées pour les textures planes, notamment celle de [WILL 83] qui est la plus simple, peuvent être généralisées au cas 3D. La structure pyramidale de [WILL 83], appelé "mip map", contient différentes versions préfiltrées de la textures (cf. paragraphe V.3.3.3). Les versions préfiltrées sont des carrés de définitions $2^n x 2^n$, $2^{n-1} x 2^{n-1}$, ..., 2x2 et 1x1 ; chaque pixel d'une version préfiltrée est le résultat d'une moyenne simple des quatre pixels correspondants sur la version de définition immédiatement supérieure. Dans le cas 3D, les version préfiltrées sont des cubes de définitions $2^n x 2^n x 2^n$, $2^{n-1} x 2^{n-1} x 2^{n-1}$, ..., 2x2x2 et 1x1x1 ; chaque voxel d'une version préfiltrée est le résultat d'une moyenne simple des huit voxels correspondant sur la version de définition immédiatement supérieure. Les interpolations bilinéaires nécessaires dans le cas 2D deviennent trilinéaires pour les textures 3D.

D'autres méthodes d'antialiassage envisageables sont celles qui tiennent compte de la technique de génération des textures procédurales. Par exemple, si la méthode de génération permet de connaître les fréquences de la textures, on peut appliquer directement l'antialiassage en atténuant ou supprimant les composantes non représentables comme dans le cas 2D [NORT 82] [GHAZ 85]. Etant donné que dans le cas de textures 3D, l'antialiassage peut être effectué indépendamment de la surface, ces méthodes peuvent être utilisées plus facilement que dans le cas précédent.

Références du chapitre V

[BERG 79] : M. BERGER,
Géométrie- 1 : Actions de groupes,espaces affines et projectifs, CEDIC/NATHAN, Avril 1979, pp. 158-159.

[BLIN 76] : J. BLINN and M.E. NEWELL,
"Texture and Reflection in Computer Generated Images", Communications of the ACM, vol. 19, No. 10, October 1976, pp. 542-547.

[CARE 87] : J. CAREY and D. GREENBERG,
"Texture for Realistic Image Synthesis", Computer and Graphics, vol. 11, pp. 73-85, 1987.

[CATM 80] : E. CATMULL and A.R. SMITH,
"3-D Transformations of Images in Scanline Order", Computer Graphics, vol. 14, No. 3, July 1980, pp. 279-285.

[CATM 74] : E. CATMULL,
"A Subdivision Algorithm for Computer Display Curved Surfaces", Ph.D. thesis, University of Utah, December 1974.

[CROW 84] : F. CROW,
"Summed-Area Tables for Texture Mapping", Computer Graphics, vol. 18, No. 3, July 1984, pp. 207-212.

[FEIB 80] : E. FEIBUSH, M. LEVOY and R. COOK,
"Synthetic Texturing Using Digital Filters", Computer Graphics, vol. 14, No. 3, July 1980, pp. 295-301.

[GAGA 83] : A. GAGALOWICZ,
Vers un modèle de textures, Thèse de Doctorat d'Etat, Université Pierre et Marie Curie, Paris VI, Mai 1983, pp. 295-301.

[GANG 82]: M. GANGNET, D. PERNY and P. COUEIGNOUX,
"Perspective Mapping of Planar Textures", Eurographics 1982, North-Holland, 1982, pp. 57-71.

[GANG 84] : M. GANGNET et D.GHAZANFARPOUR,
"Comparaison de techniques de mise en perspective de textures planes", Premier colloque international d'images électroniques, CESTA, Biarritz, Mai 1984, pp. 29-35.

[GARD 84] : G. GARDNER,
"Simulation of Natural Scenes Using Textured Quadric Surfaces, Computer Graphics, vol. 18, No. 3, July 1984, pp. 11-20.

[GARD 85] : G. GARDNER,
"Visual Simulation of Clouds", Computer Graphics, vol. 19, No. 3, July 1985, pp. 297-303.

[GHAZ 85] : D. GHAZANFARPOUR,
"Synthèse d'images et antialiassage", Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure des Mines, Saint-Etienne, novembre 1985.

[GLAS 86] : A. GLASSNER,
"Adaptative Precision in Texture Mapping", Computer Graphics, vol. 20, No. 4, August 1986, pp. 297-306;

[HECK 86] : P. HECKBERT,
"Filtering by Repeated Integration", Computer Graphics, vol. 20, No. 4, August 1986, pp. 315-321.

[JAYA 79] : S. JAYARAMAMURTHY,
"Multilevel Array Grammars for Generation Texture Scenes", Proc. IEEE, Chicago, August 1979, pp. 391-398.

[MAND 82] : B. MANDELBROT,
The Fractal Geometry of Nature, W.H. FREEMAN and Company, 1982.

[MAST 87] : G. MASTIN, A. WTTERBERG and J. MAREDA,
"Fourier Synthesis of Ocean Scenes", IEEE, Computer Graphics and Applications, March 1987, pp. 16-23.

[MILL 86] : G. MILLER,
"The Definition and Rendering of Terrain Maps", Computer Graphics, vol. 21, No. 4, August 1986, pp. 39-47.

[MITC 87] : D. MITCHELL,
"Generating Antialiased Images at Low Sampling Densities", Computer Graphics, vol. 21, No. 4, August 1987, pp. 65-72.

[MONN 81] : J. MONNE, F. SCHMITT AND D. MASSALOUX,
"Bidimensional Texture Synthesis by Markov Chains", Computer Graphics and Image Processing, vol. 17, 1981.

[NORT 82] : A. NORTON, A. ROCKWOOD and P.SKOLMOSKI,
"Clamping : A Method of Antialiasing Textured Surfaces by Bandwidth Limiting in Object Space", Computer Graphics, vol. 16, No. 3, July 1982, pp. 1-8.

[OKA 87] : M. OKA, K. TSUTSUI, A. OHBA, Y. KURAUCHI and T. TAGO,
"Real-Time Manipulation of Texture-Mapped Surfaces" Computer Graphics, vol. 21, No. 4, July 1987, pp. 181-188.

[PEAC 85] : D. PEACHEY,
"Solid Texturing of Complex Surfaces", Computer Graphics, vol. 19, No. 3, July 1985, pp. 279-286.

[PERL 85] : K. PERLIN,
"An Image Synthesizer", Computer Graphics, vol. 19, No. 3, July 1985, pp. 287-296.

[PERL 89] : K. PERLIN and E.M. HOFFERT,
"Hypertexture", Computer Graphics, vol. 23, No. 3, July 1989, pp. 253-262.

[SCHA 80] : B. SCHACHTER,
"Long Crested Wave Models", Computer Graphics and Image Processing, vol. 12, 1980, pp. 187-201.

[SMIT 84] : A. SMITH,

"Plants, Fractals and Formal Languages", Computer Graphics, vol. 18, No. 3, July 1984, pp. 1-10.

[SMIT 87] : A. SMITH,
"Planner 2-Pass Texture Mapping and Warping", Computer Graphics, vol. 21, No. 4, July 1987, pp. 263-272.

[WILL 83] : L.WILLIAMS,
"Pyramidal Parametrics", Computer Graphics, vol. 17, No. 3, July 1983, pp. 1-11.

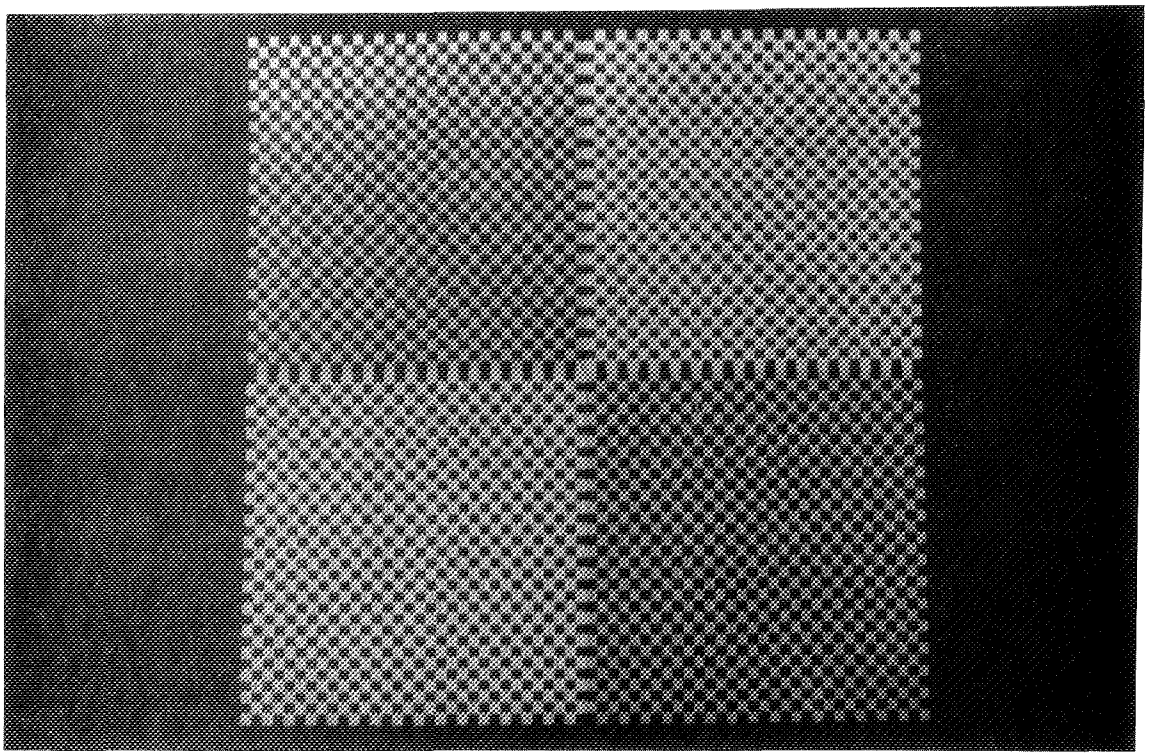


Image V.1 : Texture originale (structurée et hautes fréquences).

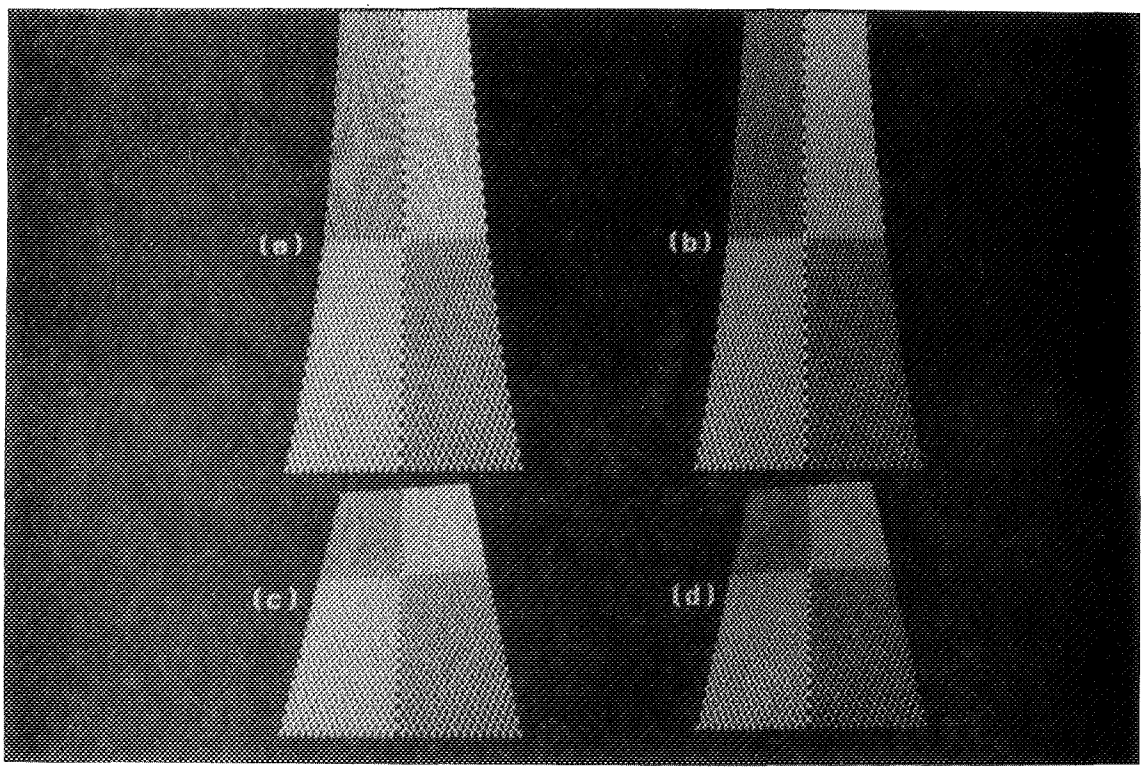


Image V.2 : Texture mise en perspective en deux passes.

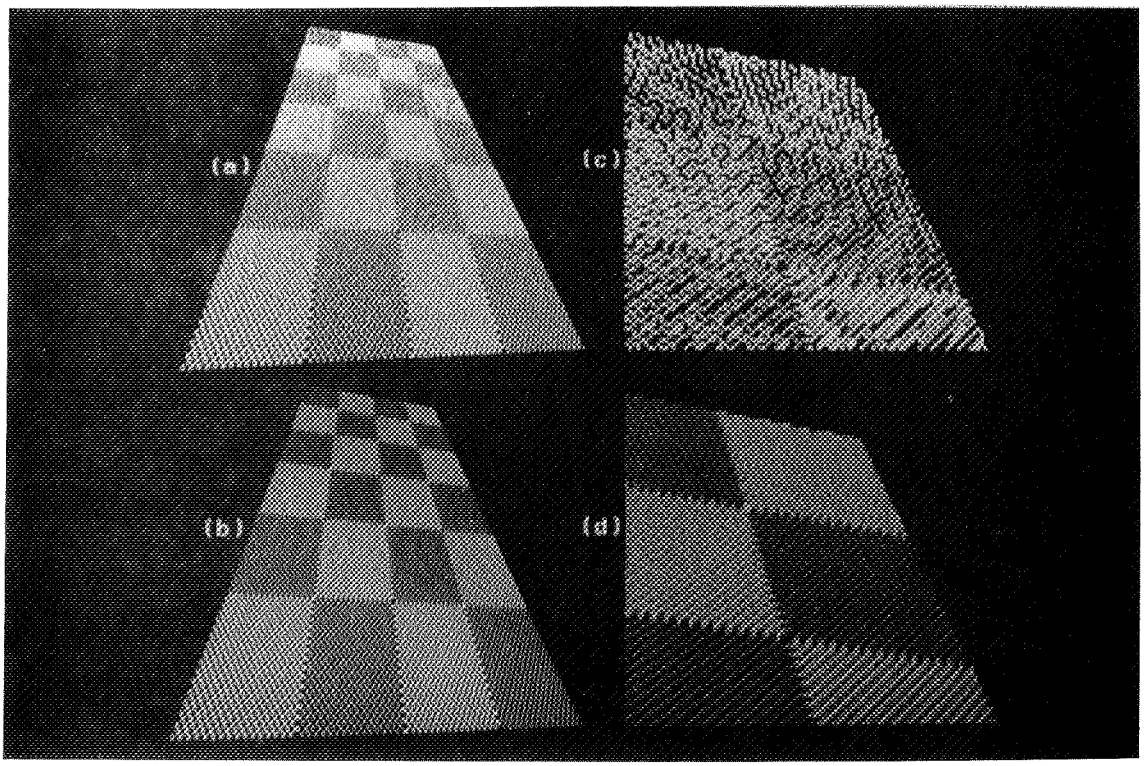


Image V.3 : Mise en perspective de texture avec la méthode de transformation directe et convolution précise.

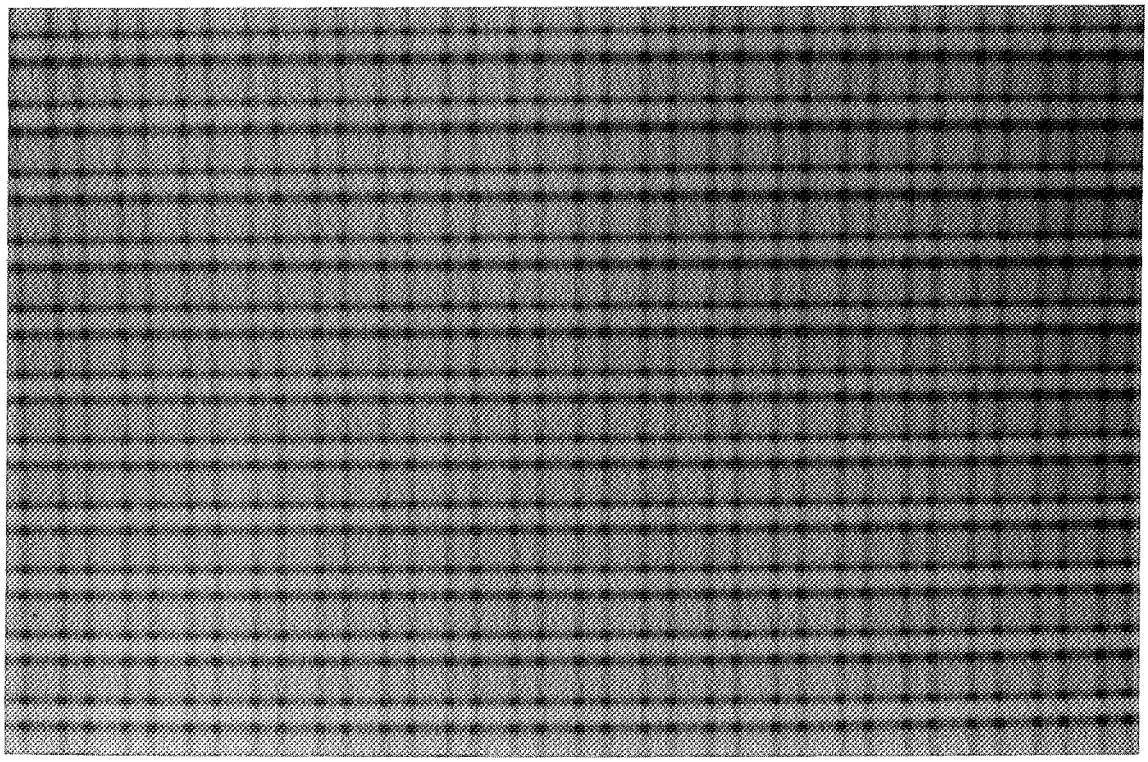


Image V.4 : Texture originale (structurée et hautes fréquences).

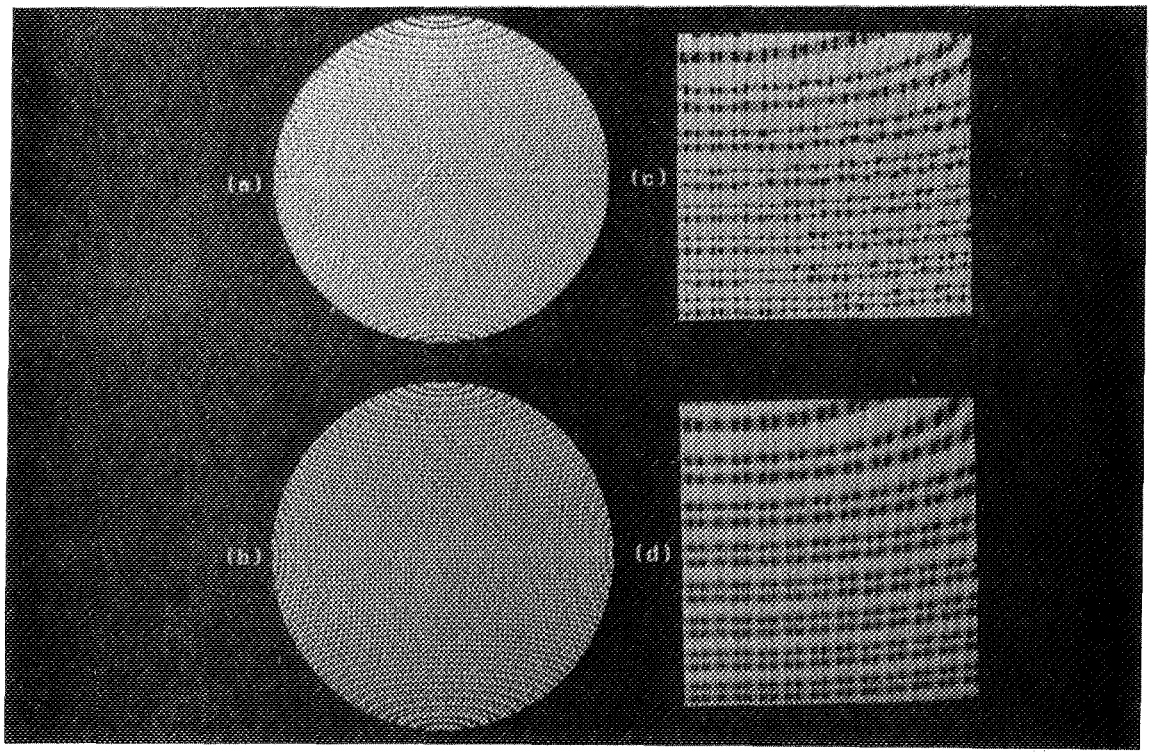


Image V.5 : Mise en perspective de texture avec la méthode de transformation directe et convolution précise.

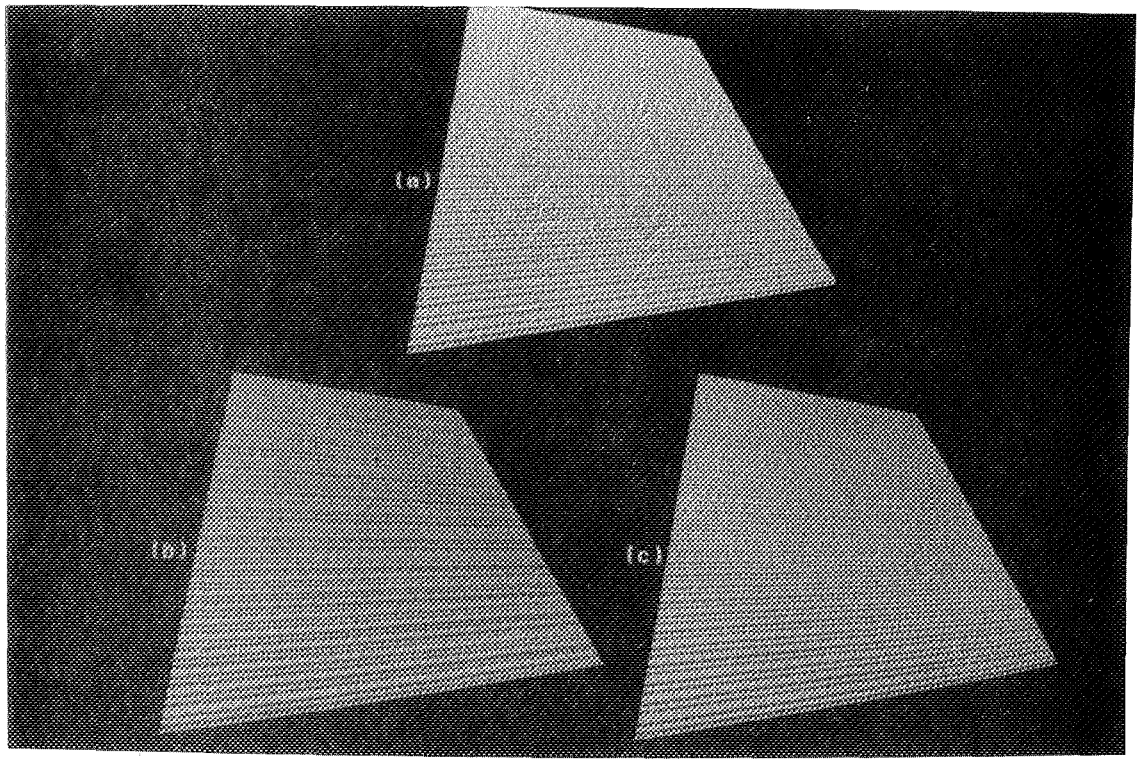


Image V.6 : Mise en perspective avec deux taux de compression très différents selon u et v ($C_u >> C_v$).

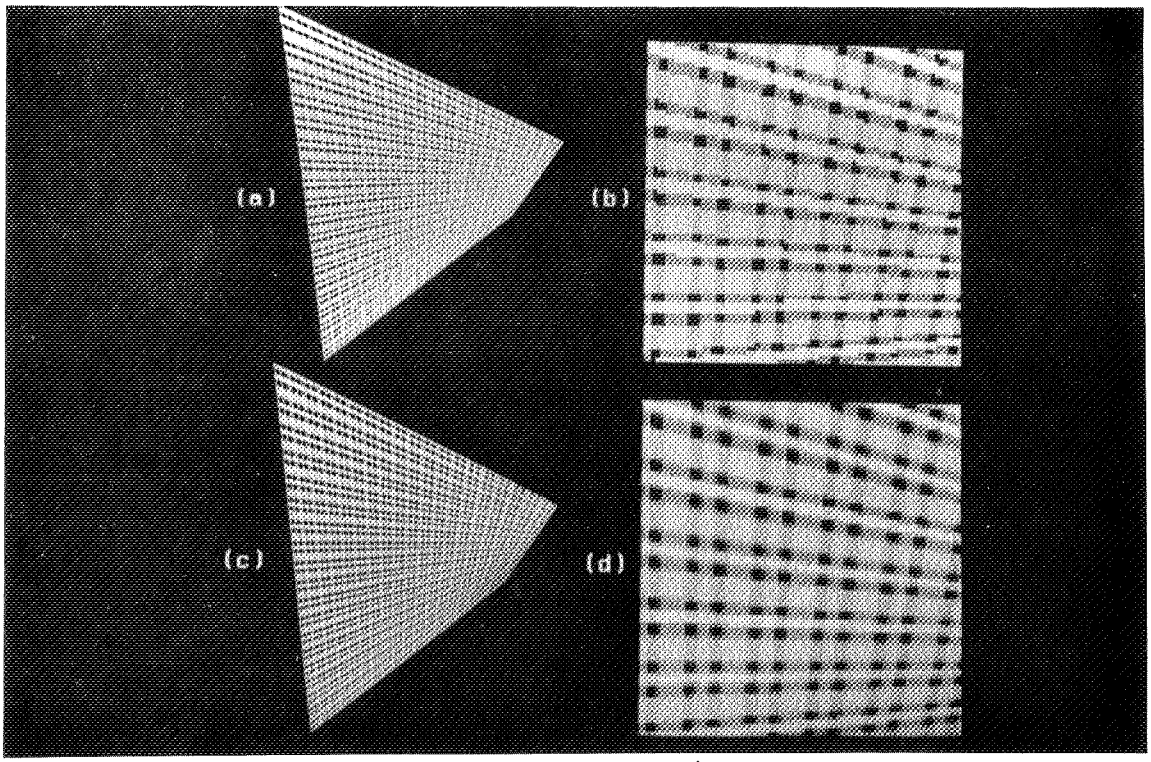


Image V.7 : Mise en perspective avec la méthode de filtrage asymétrique.

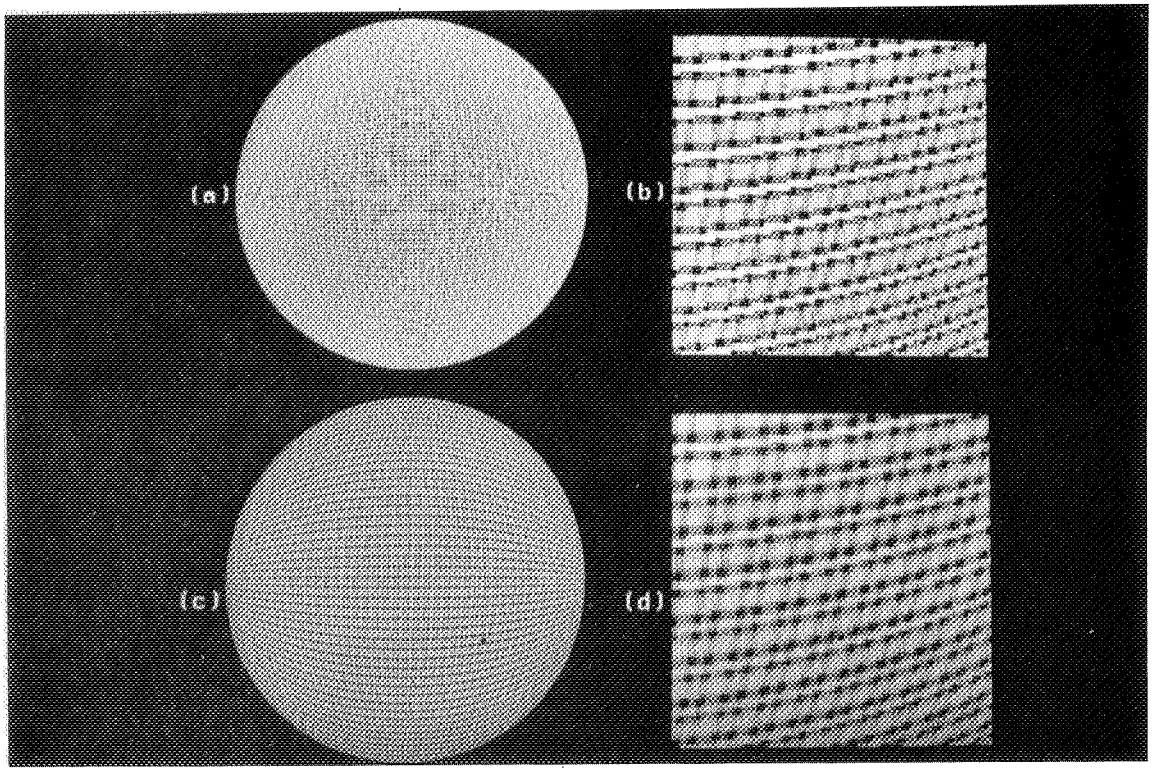


Image V.8 : Mise en perspective avec la méthode de filtrage asymétrique.

Conclusion

Conclusion

Le support de visualisation utilisé dans le cas d'images numériques est presque toujours un moniteur constitué d'un tube à rayons cathodiques. Un écran de visualisation d'un tel type nécessite la conversion numérique-analogique de l'image pour son affichage au rythme du balayage vidéo. Les contraintes technologiques liées essentiellement à la vitesse de conversion imposent une faible définition aux mémoires de trames. Il en résulte des problèmes de discrétisation dus à un échantillonnage inapproprié de l'image initialement définie sous une forme analogique.

Les problèmes de discrétisation d'images se manifestent sous forme de défauts d'aliassage. L'aliassage en synthèse d'images numériques peut être évité ou au moins sensiblement estompé à l'aide de techniques appropriées. La solution la plus courante est le filtrage passe-bas de la scène afin de supprimer ou de diminuer les hautes fréquences non représentables avant son échantillonnage. Les méthodes d'antialiassage sont différentes suivant l'algorithme de visualisation utilisé et la qualité exigée.

Le sur-échantillonnage est la technique d'antialiassage la plus générale mais la plus coûteuse en temps de calculs. En particulier, si le sur-échantillonnage est appliqué globalement, notamment dans le cas des algorithmes coûteux de visualisation, les temps de calculs sont très importants. En contrepartie, c'est une technique très efficace pour les manifestations les plus courantes de l'aliassage en synthèse d'images (marches d'escalier, par exemple). De plus, il n'existe pas de cas particuliers comme dans d'autres méthodes. Le sur-échantillonnage est bien adapté aux algorithmes de visualisation qui procèdent par balayage de ligne ou ceux utilisant une subdivision récursive de l'image.

Les méthodes de filtrage analytiques basées sur les propriétés géométriques des entités constituant la scène sont souvent utilisées avec des algorithmes incrémentaux pour l'antialiassage des contours de scènes polygonales. Contrairement à la technique de sur-échantillonnage, ces méthodes sont plutôt rapides, mais il peut exister des cas particulier tels que les sommets de polygones. Ces techniques sont bien adaptées aux algorithmes d'affichage qui procèdent polygone par polygone. Etant donnée leur simplicité, l'implantation matérielle de certaines sont envisageables.

L'échantillonnage stochastique est une nouvelle technique d'antialiassage en synthèse d'images. L'échantillonnage stochastique utilisant une fonction de type poissonien permet de remplacer l'aliassage qui est un défaut régulier par le bruit qui est un défaut irrégulier et donc moins perceptible. Cette technique est essentiellement utilisée pour la visualisation par la méthode du lancer de rayons. L'échantillonnage stochastique est souvent utilisé avec un certain degré de sur-échantillonnage.

L'application des méthodes d'antialiassage avec certains algorithmes de visualisation pose des problèmes. C'est en particulier le cas du tampon de profondeur pour lequel ces problèmes sont nombreux et difficiles à résoudre. Dans ce cas, il faut chercher des méthodes d'antialiassage qui préservent la simplicité de la technique du tampon de profondeur. De plus, la possibilité de l'implantation matérielle de ces méthodes est importante.

La présence de textures dans les images de synthèse peut provoquer le phénomène d'aliassage sous la forme de moirés. Les différentes méthodes de filtrages simultanés ou a priori avec des transformations perspectives directes ou inverses sont envisageables. Le filtrage appliqué doit être adaptatif en fonction de la compression de la texture afin d'éviter les problèmes des parties floues dues à un sur-filtrage ou la présence de moirés dans les parties sous-filtrées de l'image.

ANNEXE A

Echantillonnage régulier, aliassage et antialiassage d'un signal monodimensionnel

ANNEXE A

Echantillonnage régulier, aliassage et antialiassage d'un signal monodimensionnel

A.1 Echantillonnage régulier et aliassage

Le résultat fondamental est le théorème d'échantillonnage et de reconstitution d'un signal analogique (théorème de Shannon). Etant donnée son importance, nous rappelons ci-dessous ce résultat.

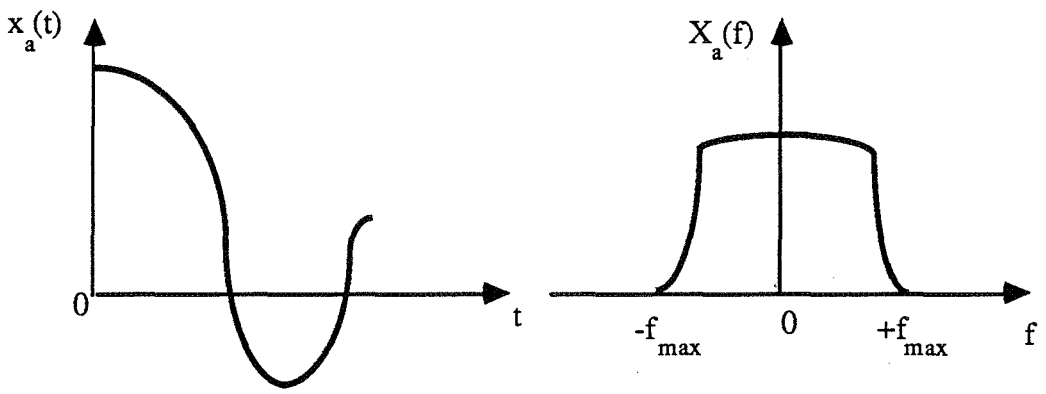
Un signal analogique monodimensionnel est un signal dont l'amplitude varie de façon continue en fonction d'une variable, par exemple le temps. Soit $x_a(t)$ un signal analogique à support non-borné dans le domaine du temps (cf.figure A.1.a). On définit la transformée de Fourier de $x_a(t)$ par :

$$X_a(f)= \int_{-\infty}^{+\infty} x_a(t).e^{-j2\pi ft}.dt \qquad (A.1)$$

Elle définit le spectre fréquentiel du signal (cf.figure A.1.b). $x_a(t)$ étant un signal à support non-borné dans le domaine du temps, $X_a(f)$ est un signal à support borné (on dit encore à bande limitée) dans le domaine fréquentiel :

$$X_a(f)= 0 \quad \text{si} \quad |f|>f_{max} \qquad (A.2)$$

où f_{max} est la fréquence maximale que contient le signal analogique. En pratique les signaux physiques n'ont pas une bande limitée : la largeur de bande d'un signal est déterminée par le fait qu'au delà d'une certaine fréquence, le pourcentage d'énergie contenue dans le spectre est négligeable par rapport à l'énergie totale.



(a) signal analogique (b) spectre du signal analogique

Figure A.1 : Représentations temporelle et fréquentielle d'un signal analogique monodimensionnel.

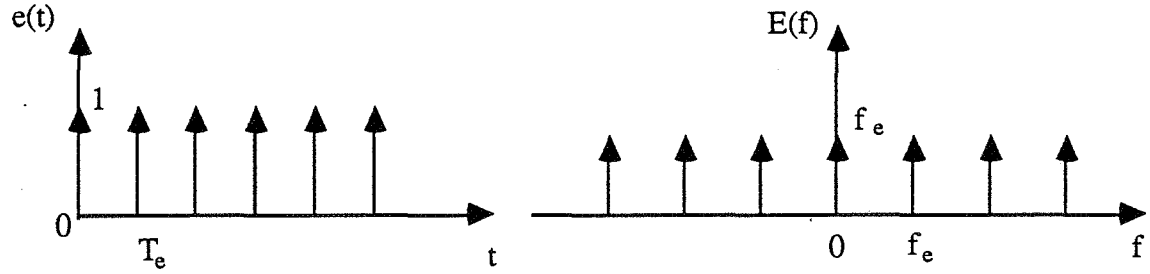
Le signal d'échantillonnage est une suite périodique d'impulsions de Dirac (cf.figure A.2.a) définie par :

$$e(t)=\sum_{n=-\infty}^{n=+\infty}\delta(t-n.T_e) \qquad (A.3)$$

où δ est la distribution de Dirac définie par : $\delta(t-t_0)=1$ si $t=t_0$ et zéro partout ailleurs et T_e est la période d'échantillonnage. La représentation de e(t) dans le domaine fréquentiel est donnée par sa transformée de Fourier :

$$E(f)=f_e\sum_{n=-\infty}^{n=+\infty}\delta(f-n.f_e) \qquad (A.4)$$

où $f_e=\frac{1}{T_e}$ est la fréquence d'échantillonnage (cf.figure A.2.b).



(a) signal d'échantillonnage (b) spectre du signal d'échantillonnage

Figure A.2 : Représentations temporelle et fréquentielle d'un signal d'échantillonnage monodimensionnel.

Le signal échantillonné $x_e(t)$ est défini par un produit (ordinaire) dans le domaine temporel :

$$x_e(t)= x_a(t).e(t)= x_a(t) \sum_{n=-\infty}^{n=+\infty} \delta(t-n.T_e) \qquad (A.5)$$

ou par un produit de convolution, noté $*$, dans le domaine fréquentiel :

$$X_e(f)= X_a(f)*E(f)= f_e \sum_{n=-\infty}^{n=+\infty} X_a(f-n.f_e) \qquad (A.6)$$

$X_e(f)$ est un signal périodique dans le domaine fréquentiel de période f_e. La figure A.3.a représente le signal échantillonné et les figures A.3.b, A.3.c et A.3.d le spectre de ce signal dans les trois cas possibles : $f_e>2.f_{max}$, $f_e=2.f_{max}$ et $f_e<2.f_{max}$.

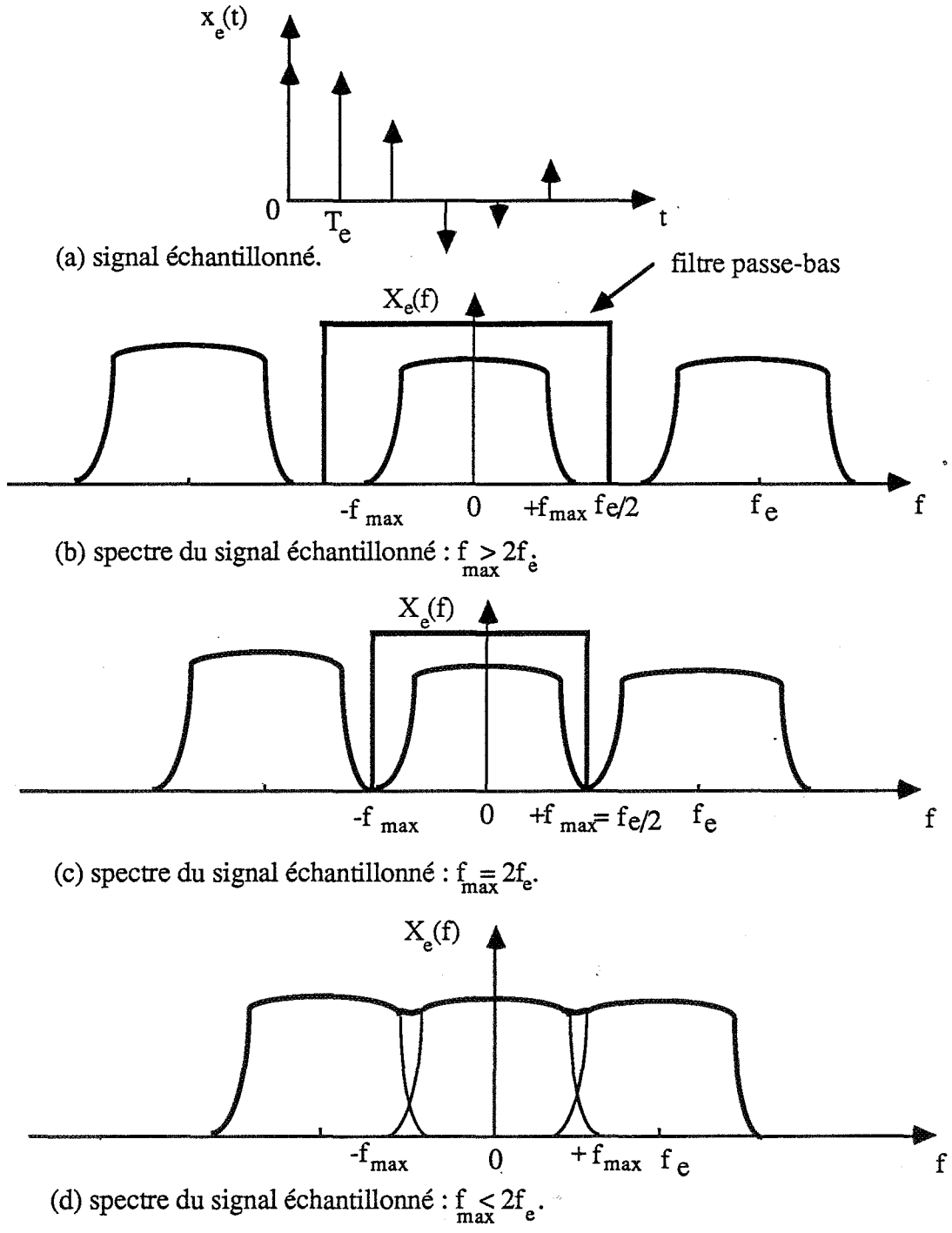


Figure A.3 : Le signal échantillonné et les trois cas possibles pour son spectre.

On constate, dans chaque cas, que la restriction de $X_e(f)$ à un domaine correspondant à une période (appelée le spectre secondaire du signal) est proportionnelle au spectre $X_a(f)$ du signal analogique de départ (cf. la relation A.6). Considérons successivement les trois cas :

Premier cas : fe>2.fmax

Dans ce cas, les supports des motifs du spectre du signal échantillonné sont disjoints (cf. figure A.3.b). Montrons que, dans ce cas, le signal analogique d'origine, $x_a(t)$, peut être reconstitué fidèlement en gardant le motif central du spectre et en supprimant tous les motifs voisins ; ceci correspond à un filtrage passe-bas. On a en effet :

$$X_a(f)=X_e(f).H(f) \tag{A.7}$$

où H(f) est la fonction de transfert du filtre passe-bas idéal appelé filtre de reconstitution (cf. figure A.4.a). Pour ce filtre passe-bas idéal de fréquence de coupure $\frac{f_e}{2}$ nous avons :

$$H(f) = \begin{cases} \frac{1}{f_e} & \text{si } |f| \le \frac{f_e}{2} \\ 0 & \text{si } |f| > \frac{f_e}{2} \end{cases} \tag{A.8}$$

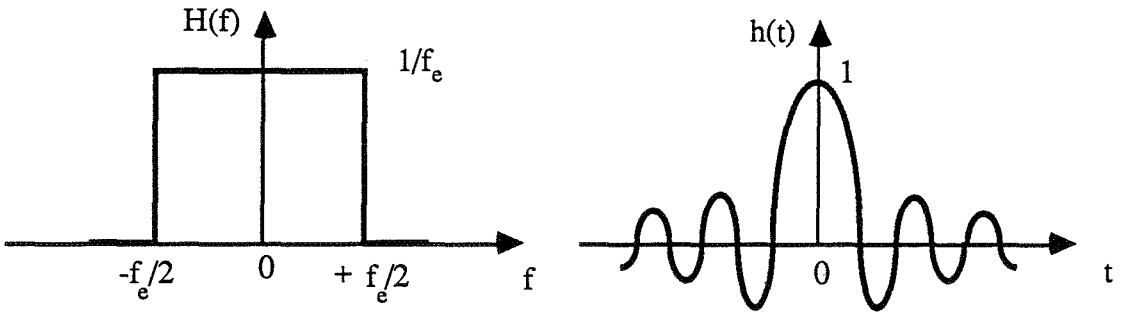
L'opération de filtrage dans le domaine temporel est équivalente au produit de convolution suivant :

$$x_a(t)= x_e(t)*h(t) \tag{A.9}$$

où h(t) est la réponse impulsionnelle du filtre passe-bas idéal (cf. figure A.4.b), qui est la transformée de Fourier inverse de H(f) :

$$h(t)= \int_{n=-\infty}^{n=+\infty} H(f).e^{j2\pi ft}.df = \frac{1}{f_e} \int_{n=-\frac{f_e}{2}}^{n=+\frac{f_e}{2}} e^{j2\pi ft}.df$$

$$= \frac{\sin\left(\pi \cdot \frac{t}{T_e}\right)}{\pi \cdot \frac{t}{T_e}} = \mathrm{Sinc}\left(\frac{t}{T_e}\right) \tag{A.10}$$



(a) fonction de transfert (b) réponse impulsionnelle

Figure A.4 : Filtre passe-bas idéal dans le cas monodimensionnel.

Le signal analogique $x_a(t)$ est obtenu par :

$$x_a(t)= \sum_{n=-\infty}^{n=+\infty} x(n.T_e).\mathrm{Sinc}\left(\frac{t-n.T_e}{T_e}\right) \qquad (A.11)$$

Deuxième cas : $f_e=2.f_{max}$

Dans ce cas, les supports de deux motifs consécutifs du spectre du signal échantillonné ont un seul point commun (cf.figure A.3.c). Le signal analogique $x_a(t)$ peut être théoriquement reconstitué fidèlement à partir de ses échantillons à l'aide d'un filtre passe-bas idéal de fréquence de coupure $\frac{f_e}{2}$.

Troisième cas : $fe<2.fmax$

Dans ce cas, les supports de deux motifs consécutifs du spectre du signal échantillonné se recouvrent (cf.figure A.3.d). Ce phénomène de repliement du spectre s'appelle *l'aliassage*. La reconstitution fidèle du signal analogique $x_a(t)$ à partir du signal sous-échantillonné $x_e(t)$ n'est plus possible : si on essaie de reconstituer le signal analogique $x_a(t)$ à l'aide d'un filtre passe-bas de fréquence de coupure égale à $\frac{f_e}{2}$ par exemple, les composantes hautes fréquences du motif central recoupent le motif voisin dans l'intervalle $[fe-f_{max}, \frac{f_e}{2}]$ du spectre du signal analogique restitué, $X'_a(f)$. L'effet produit est une perturbation des hautes fréquences du signal restitué par rapport au signal initial.

De ce qui précède, on peut déduire :

Théorème d'échantillonnage :

Un signal analogique à largeur de bande finie peut être reconstitué fidèlement à partir d'une suite complète de ses échantillons à condition que la fréquence d'échantillonnage soit au moins égale à deux fois la fréquence maximale du signal analogique, [SHAN 49]. La fréquence minimale autorisée pour l'échantillonnage correct d'un signal analogique est souvent appelée la fréquence de Nyquist.

L'interpolation correcte pour la reconstitution du signal analogique de départ (cf. la relation A.11) est obtenue à l'aide du filtre passe-bas idéal (cf. figure A.4). Mais en pratique le filtre passe-bas idéal n'existe pas ; en fait, la fonction de transfert d'un filtre passe-bas réel n'est pas rectangulaire (cf. figure A.5). La restitution correcte d'un signal analogique par un tel filtre n'est envisageable que si une bande de garde entre les motifs voisins du spectre du signal échantillonné existe. Il est donc préférable d'avoir une fréquence d'échantillonnage supérieure à la fréquence de Nyquist (signal sur-échantillonné) pour avoir des motifs du spectre du signal échantillonné disjoints (cf.figure A.3.b), au lieu d'une fréquence de Nyquist (cf. figure A.3.c). Par exemple, dans les systèmes standard de téléphone, on échantillonne à 8 KHz le signal de parole utilisé alors que celui-ci a une largeur de bande limitée à 3 KHz environ.

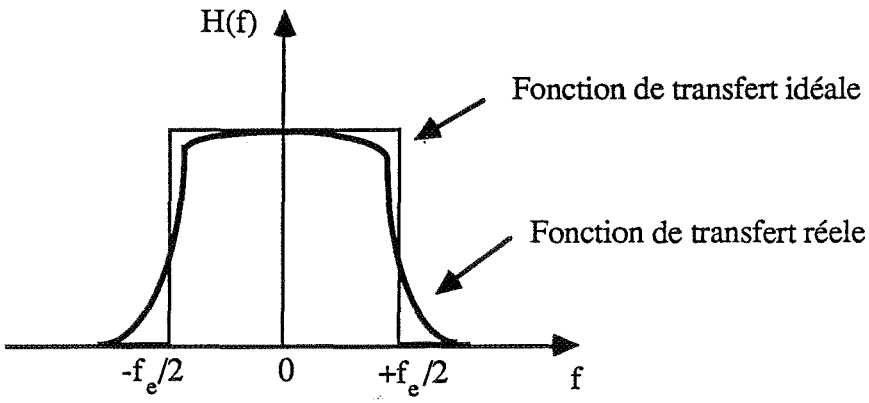


Figure A.5 : Fonction de transfert d'un filtre passe-bas réel.

Pour terminer ce paragraphe, nous pouvons donner un exemple bien connu du phénomène étudié.

Exemple :

Une caméra de cinéma prend 24 images/s . Dans la scène à filmer il existe un objet qui tourne dans un certain sens à une fréquence supérieure à 12 cycles/s ; le résultat est, en général, un objet qui semble tourner soit dans l'autre sens, soit assez lentement dans le même sens. Les 24 échantillons par seconde ne sont donc pas suffisants pour un objet qui tourne aussi rapidement.

A.2 Antialiassage et reconstitution

L'ampleur du phénomène d'aliassage (repliement du spectre) dépend de la quantité d'énergie située dans la zone aliassée (cf.figure A.3.d) par rapport à l'énergie totale du signal. Quelle que soit cette ampleur, l'aliassage est un phénomène irréversible par rapport au signal échantillonné. Il faut donc prendre des précautions avant d'échantillonner le signal analogique. Pour obtenir un échantillonnage correct, deux solutions sont envisageables :

1- Augmentation de la fréquence d'échantillonnage :

Cette solution, outre les contraintes technologiques qu'elle impose, en particulier, pour les images numériques (cf. chapitre II), peut être très coûteuse dans certains cas : en effet, si la fréquence maximale du signal analogique est relativement élevée, nous aurons besoin alors d'un nombre important d'échantillons. De plus, les signaux physiques n'ont pas une largeur de bande finie ; en fait, on suppose que le pourcentage d'énergie d'un signal à partir d'une certaine fréquence est négligeable par rapport à son énergie totale (cf.figure A.6). Théoriquement, l'échantillonnage des signaux physiques ne sera donc jamais correct, quelle que soit la fréquence d'échantillonnage utilisée.

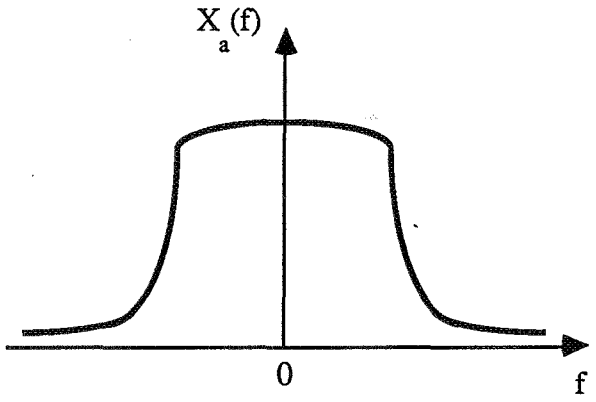


Figure A.6 :Spectre d'un signal physique.

2- Limitation du spectre fréquentiel du signal analogique :

Cette solution consiste à limiter le spectre fréquentiel du signal analogique avant l'échantillonnage en utilisant un filtre passe-bas appelé filtre antirepliement.

Soient :

- $x'_a(t)$ un signal analogique temporel et $X'_a(f)$ son spectre fréquentiel,
- $h'(t)$ la réponse impulsionnelle et $H'(f)$ la fonction de transfert du filtre passe-bas antirepliement :

$$H'(f) = \begin{cases} 1 & \text{si } |f| \leq f_c \\ 0 & \text{si } |f| > f_c \end{cases}$$

Le résultat du filtrage est un signal analogique dont le support fréquentiel est limité à $|f_c| \leq f_e/2$, (cf figure A.7). Notons $x_a(t)$ (resp. $X_a(f)$) la représentation temporelle (resp. fréquentielle) de ce signal. L'opération de filtrage est définie par :

$$X_a(f) = X'_a(f).H'(f) \qquad \text{(A.12)}$$

ou directement par un produit de convolution dans le domaine temporel :

$$x_a(t) = h'(t) * x'_a(t) \qquad \text{(A.13)}$$

où $x_a(t)$ est un signal à support non-borné dans le domaine temporel.

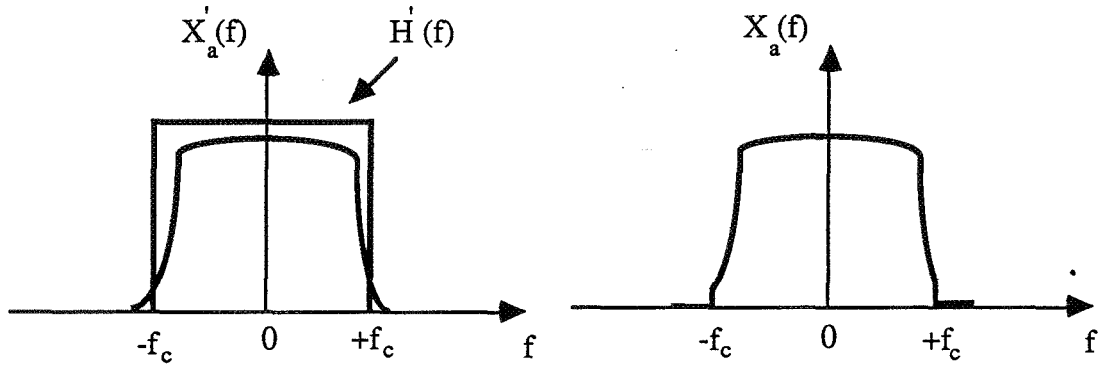


Figure A.7 : Limitation du spectre fréquentiel.

A.3 Chaîne d'échantillonnage et de reconstitution idéale

La figure A.8 montre une chaîne d'échantillonnage et de reconstitution idéale, mais sans module de traitement des échantillons. Cette chaîne est composée de trois modules :

1- Un premier filtre passe-bas idéal de fréquence de coupure $f_c \leq \frac{f_e}{2}$; c'est le filtre antirepliement.

2- Un multiplicateur qui est l'échantillonneur.

3- Un deuxième filtre passe-bas idéal de fréquence de coupure $\frac{f_e}{2}$; c'est le filtre de reconstitution qui a un rôle d'interpolation idéale. Il est évident que cette chaîne n'a d'intérêt que si des traitements sont effectués sur le signal échantillonné.

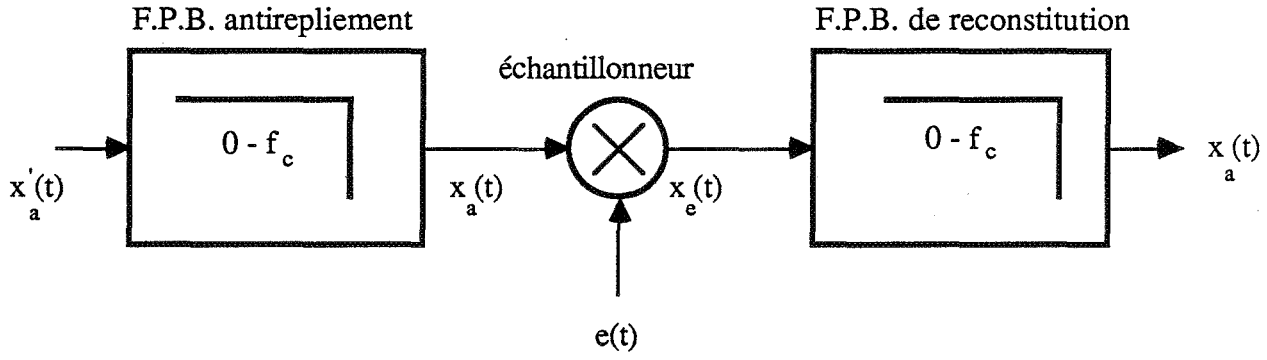


Figure A.8 : Une chaîne d'échantillonnage et de reconstitution idéale.

Pour d'autres informations sur la théorie de l'échantillonnage on peut consulter [COUL 84] et [OPPE 75].

Remarque :

Bien que nous ayons développé cette discussion pour un signal monodimensionnel temporel, celle-ci reste valable pour n'importe quel signal monodimensionnel, par exemple un signal de paramètre spatial monodimensionnel.

Références de l'annexe A

[COUL 84] : F. de COULON,
Traité d'électricité, vol. VI : Théorie et traitement des signaux, Editions Georgi, 1984, pp. 273-303.

*[OPPE 75] : A. OPPENHEIM and R. SCHAFER,*Digital Signal Processing, Prentice-Hall, 1975, pp. 26-30

[SHAN 49] : C. SHANNON,
"Communication in the Presence of Noise", Proc. IRE, vol. 37, 1949, pp. 10-21.

ANNEXE B

Réponse de l'analyse d'une mire par les spots différents

ANNEXE B

Réponse de l'analyse d'une mire par les spots différents

La figure B.1 représente un ensemble de réponses d'analyse sur une mire composée de barres noires et blanches (mire de Foucault) [GOUS 72]. La largeur des barres est égale à D. Les trois premiers cas concernent un spot à base circulaire de diamètre d, ayant une distribution uniforme : premier cas d=0, deuxième cas d<D, troisième cas d>D. Le quatrième cas représente la réponse de l'analyse avec un spot non-symétrique ayant une distribution non-uniforme.

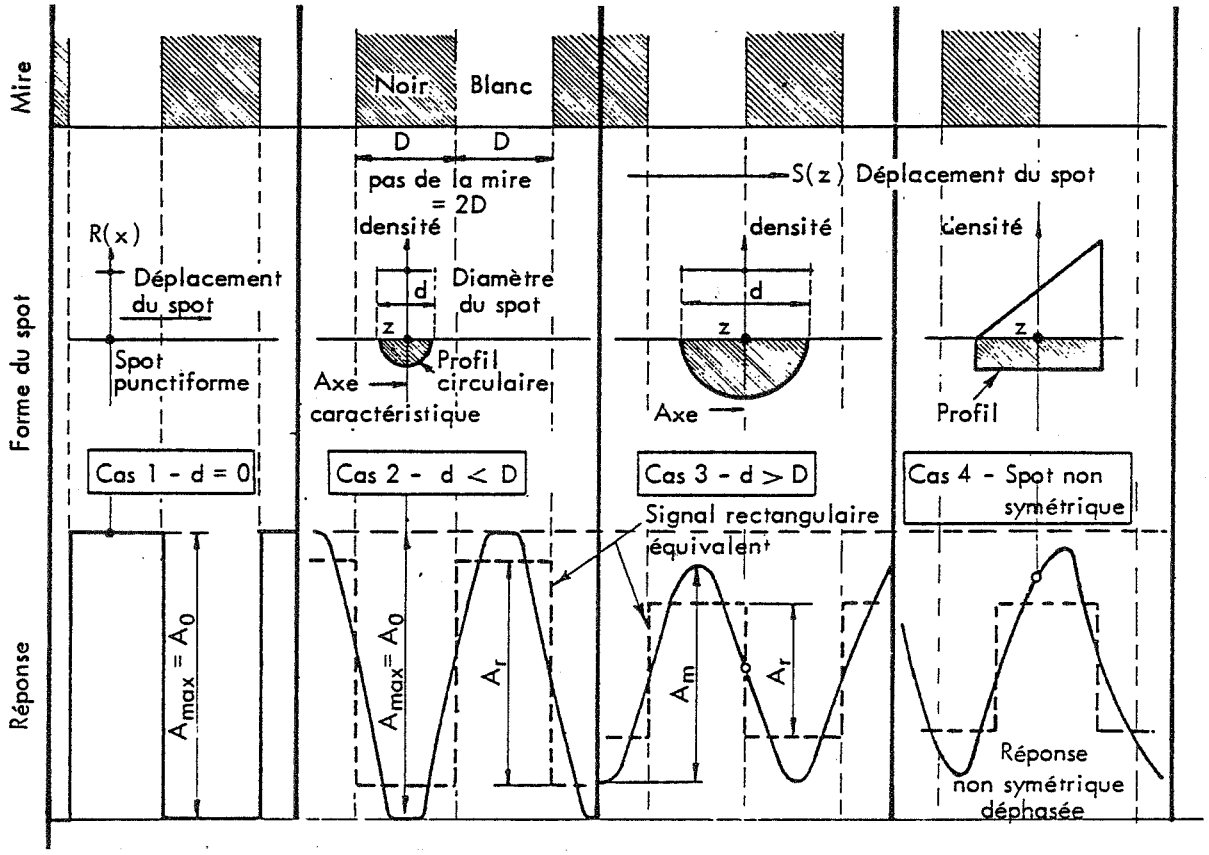


Figure B.1 : Réponse de l'analyse d'une mire de Foucault par 4 spots différents d'après Goussot [GOUS 72].

Dans le cas 1 (d=0), on parle de spot "ponctuel" ; la réponse de l'analyse est un signal rectangulaire ayant la même période spatiale que la mire. Il représente les changements d'intensité abrupts de la scène (hautes fréquences). Dans le cas 2 (d<D), la réponse de l'analyse est un signal dont les changements d'intensité sont atténués ; l'énergie contenue dans la zone de hautes fréquences est par conséquent moins importante que dans le premier cas. Dans le cas 3 (d>D), la réponse de l'analyse est un signal quasi-sinusoïdal (les harmoniques sont pratiquement supprimées), de même période spatiale que la mire. On peut dire que c'est un signal à bande limitée. Dans ce cas, le résultat est identique à celui qu'on aurait obtenu en appliquant un filtrage passe-bas sur la scène (mire) suivi d'une analyse par un spot ponctuel. Dans le cas 4, où le spot est non symétrique, la réponse de l'analyse est non symétrique et déphasée. En outre, l'énergie contenue dans la zone des hautes fréquences est moins importante que dans le premier cas. On peut en déduire que quelle que soit la forme d'un spot d'analyse réel (non ponctuel), il y aura un effet de filtrage passe-bas sur la scène.

Référence de l'annexe B

[GOUS 72] : L. GOUSSOT,

La télévision monochrome et en couleur, Edition Eyrolles, 1972, pp. 39-41.

ANNEXE C

Configurations matérielles

ANNEXE C

Configurations matérielles

Une partie des travaux présentés dans cette thèse a été réalisé avec la configuration matérielle du Centre de Recherche de l'Université Louis Pasteur de Strasbourg et une autre partie avec celle du Département d'Informatique de l'Ecole Nationale Supérieure des Mines de Saint-Etienne. Les logiciels sont développés en langage C sous un système d'exploitation Unix.

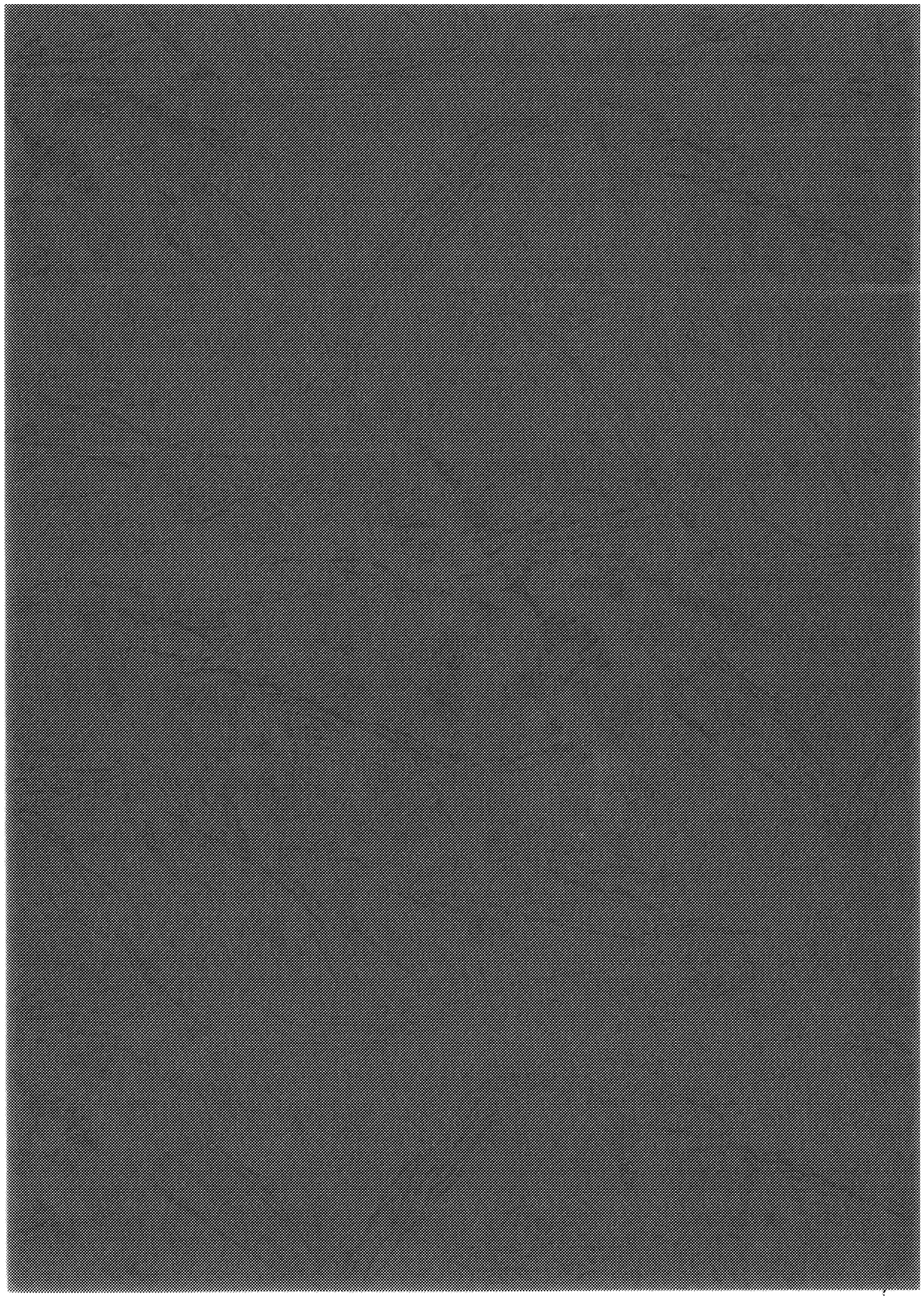
Le dispositif de saisie utilisé est un système "HAMAMATSU C10000-01" à base d'une caméra noir et blanc. La définition utilisée pour la saisie d'images est de 512x512 pixels. Une image couleur est obtenue par trois saisies avec trois filtres rouge, vert et bleu. Les particularités de ce système de saisie sont :

- un tube d'analyse vidicon ;
- une distorsion géométrique de +1% ;
- une largeur de bande nominale de 0-15 MHz ;
- un rapport nominale de signal/bruit supérieur à 40 db ;
- un convertisseur analogique-numérique de 8 bits.

Toutes les images présentées dans ce mémoire ont été affichées sur un moniteur de 19" d'un système graphique couleur de définition d'affichage de 1024x1280 pixels. Le système utilisé est une station graphique "IRIS-4D/20" avec un système d'exploitation Unix V.3. Les principaux dispositifs de ce système sont :

- un processeur RISC de 10 Mips ;
- un coprocesseur de 0,9 Mflops ;
- 8 méga octets de RAM dynamique de mémoire centrale ;
- un disque de 344 méga octets ;
- 24 plans mémoires pour le codage des couleurs ;
- une table de transcodage de couleur de 12 bits en entrée et de 3 octets en sortie ;
- un tampons de profondeur de 24 plans de mémoires ;

- 4 plans de mémoire pour "overlay/underlay" ;
- 4 plans de mémoires pour la gestion de fenêtre ;
- un moniteur couleur de 19" avec trois signaux d'entrée RVB.



Ecole Nationale Supérieure
des Mines de Saint-Etienne

N° d'ordre : 2610

Djamchid GHAZANFARPOUR-KHOLENDJANY

PROBLEMES DE DISCRETISATION ET DE FILTRAGE POUR LA VISUALISATION D'IMAGES ALGEBRIQUES

Spécialité : Informatique, Image, Intelligence Artificielle et Algorithmique

Résumé

La faible définition des mémoires de trames imposée par des contraintes technologiques pose des problèmes de discrétisation de l'image lors de son affichage. Il en résulte les différents défauts d'aliassage dus à un échantillonnage insuffisant de l'image sous sa forme analogique. Ces défauts sont perceptibles essentiellement sous formes de marches d'escalier sur les contours de l'image, d'apparition et de disparition des petits objets suivant leur position dans la scène et de présence de moirés dans les scènes portant des textures.

Pour atteindre un plus grand degré de réalisme en synthèse d'images, il est indispensable de résoudre ces problèmes de discrétisation. La solution générale est un préfiltrage passe bas de l'image avant son affichage. Nous abordons ces problèmes sous un angle théorique et pratique dans cette thèse. Nous étudions les méthodes d'antialiassage en synthèse d'images dans les cas les plus courants. Nous proposons des nouvelles méthodes en particulier des algorithmes originaux pour résoudre ces problèmes dans les cas du tampon de profondeur et des textures.

Mots-clés :

Synthèse d'images, Algorithmes de rendu, Tampons de profondeur, Textures, Discrétisation d'images, Traitement du signal, Echantillonnage régulier, Echantillonnage stochastique, Aliassage, Antialiassage, Filtrage numérique.